MAKE | BUILD | HACK | CREATE

HackSpace

TECHNOLOGY IN YOUR HANDS

hsmag.CC | June 2021 | Issue #43

3D DESIGN

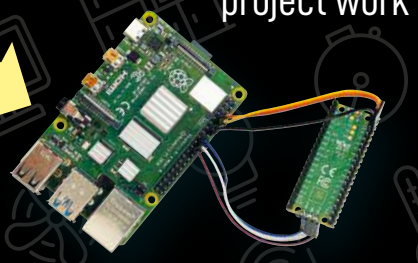
Make complex assemblies

10 INTERNET OF THINGS PROJECTS

Build your own **connected** devices

DEBUGGING

Make your microcontroller project work



MIDI CONTROLLER

Build a portable music machine

SPACE KNOTS

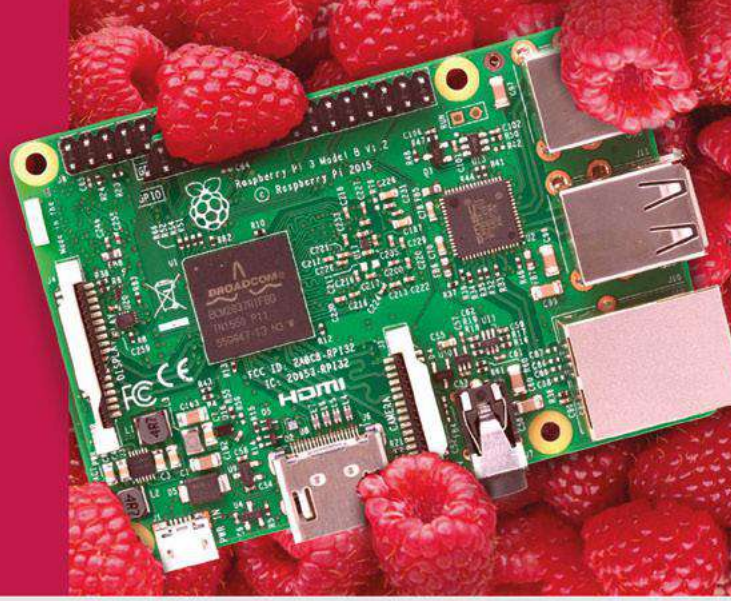
Learn how NASA ties Mars rovers together

June 2021
Issue #43 £6

9 772515 514006 43

PICO HELI-COILS CYBERDECK ROBOTS

American **Raspberry Pi** Shop



- Displays
- HATs
- Sensors
- Cases
- Arcade
- Swag
- Project Kits
- Cameras
- Power Options
- Add-on Boards
- Cables and Connectors
- GPIO and Prototyping

Partner and official reseller for top Pi brands:

adafruit · sparkfun ELECTRONICS · PIMORONI · HiFiBerry · and many others!

Price, service, design, and logistics support for **VOLUME PROJECTS**

USA PiShop.us

Canada BuyaPi.ca

Raspberry Pi APPROVED RESELLER

Welcome to HackSpace magazine

The Internet of Things (IoT) offers us a huge opportunity to automate and simplify our lives. In order for us to achieve these benefits though, we need open devices that we can connect to each other. Unfortunately, tech companies often seem more interested in selling closed devices that require us to use only their services, and only work with other devices from the same manufacturer.

Simple devices can have a big impact when they work together, and in this issue we're looking at how makers **are building their own networks of things**

This isn't a problem for makers though. With a huge range of programmable internet-enabled boards to choose from, and a mind-blowing range of sensors and actuators at our disposal, we can build our own IoT.

Simple devices can have a big impact when they work together, and in this issue we're looking at how makers are building their own networks of things to make their lives easier, happier, and more hackable.

BEN EVERARD

Editor ben.everard@raspberrypi.com

EDITORIAL

Editor
Ben Everard
ben.everard@raspberrypi.com

Features Editor
Andrew Gregory
andrew.gregory@raspberrypi.com

Sub-Editors
David Higgs, Nicola King

DESIGN

Critical Media
criticalmedia.co.uk

Head of Design
Lee Allen

Designers
Sam Ribbits, Lucy Cowan, Ty Logan

Photography
Brian O'Halloran

CONTRIBUTORS

Rosie Hattersley, Jo Hinchliffe, Marc de Vinck, Andrew Lewis, Alex Bate, Tony Goodhew, Rob Miles

PUBLISHING

Publishing Director
Russell Barnes
russell@raspberrypi.com

Advertising
Charlie Milligan
charlotte.milligan@raspberrypi.com

DISTRIBUTION

Seymour Distribution Ltd
2 East Poultry Ave,
London EC1A 9PT
+44 (0)207 429 4000

SUBSCRIPTIONS

Unit 6, The Enterprise Centre,
Kelvin Lane, Manor Royal,
Crawley, West Sussex, RH10 9PE

To subscribe
01293 312189
hsmag.cc/subscribe

Subscription queries
hackspace@subscriptionhelpline.co.uk

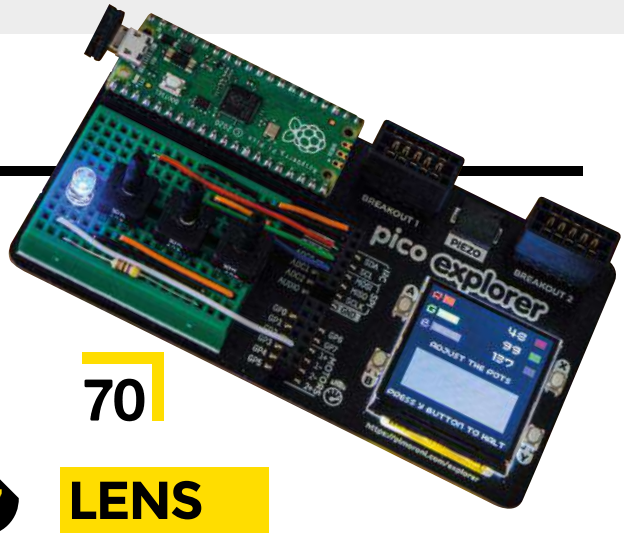
recycle

CC BY NC SA

This magazine is printed on paper sourced from sustainable forests. The printer operates an environmental management system which has been assessed as conforming to ISO 14001.

ISSN: 2515-5148.

Contents



70

06 SPARK

37 LENS

Tutorial

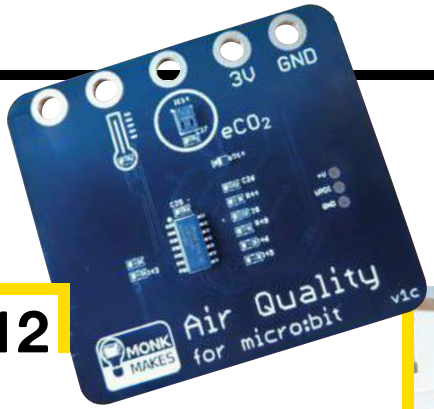
Heli-Coils

Cover Feature

MAKE THE INTERNET OF THINGS WORK FOR YOU

Big Tech is taking over – we've got to take the power back

52

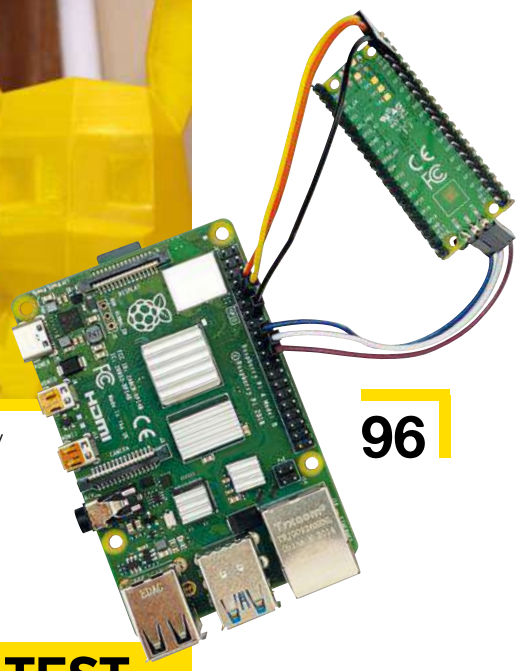


Direct from Shenzhen

Tachometer

Interview

Joel Telling

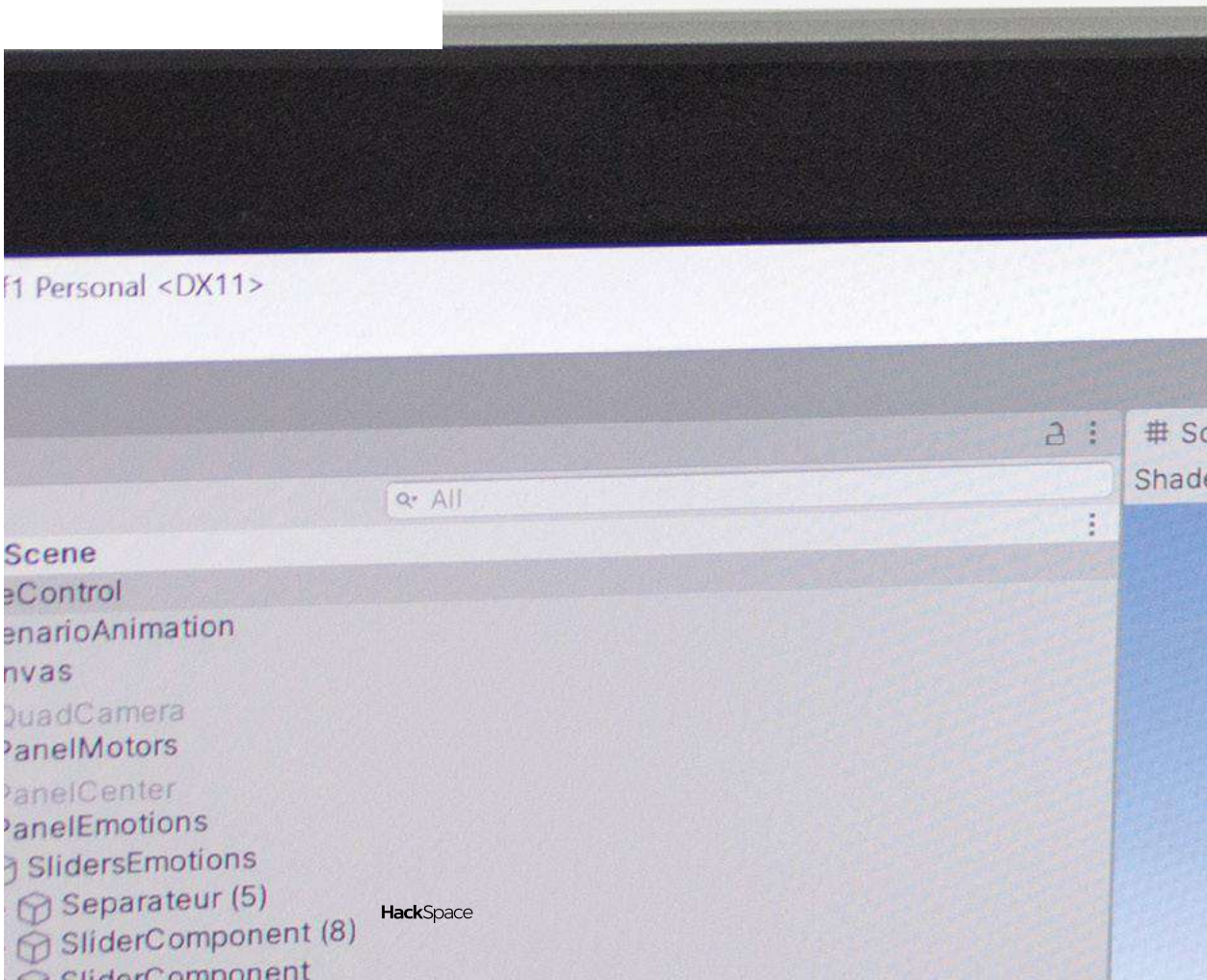


Eyecam

By **Marc Teyssier et al**

hsmag.cc/Eyecam

K, this is just odd. We explore some of the ways in which IoT makes life easier, starting on page 38; this IoT project just makes life creepier. That's the whole point: its creator, Marc Teyssier, wants to explore the human implications of being watched, all the time, by connected cameras, whether that's on Skype, Zoom, or the ever-present CCTV. The camera itself is little more than a Raspberry Pi Zero and a Camera Module. But the Eyecam reacts to motion, blinks, and holds eye contact, requiring an Arduino Nano and six servos arranged to mimic the musculature of the human eye socket. It sits in a 3D-printed skull, with silicone skin and, most horrifically of all, has human hair for eyelashes and eyebrows. Big Brother is watching you!



Right
Always in focus/ You can't feel my stare/I zoom into you/You don't know I'm there/ I take a pride in probing all your secret moves/ My tearless retina takes pictures that can prove


cene
Package Manager
ed Wiferam
2D
0

REGULAR

Linear clock

By James Wilson

hsmag.cc/LinearClock

There's a time and a place for the half-finished aesthetic. Wires everywhere, visible bits of PCB or even perfboard, and a slight roughness around the edges help give a project a homemade feel. But when you're as good at woodwork as James Wilson is, why not go the extra mile and make your projects beautiful?

Look closer at the display: there's something not right about it. Rather than break up the day into 24 hours, James has decided to display time as a percentage of the day that's elapsed. Even better than that, the display actually marks the percentage of daylight that has elapsed, with the blue dot showing the middle point of the day when the sun is at its highest. That's a lot to take in, and a lot for the electronics to take care of. James borrowed the calculations from *Astronomical Algorithms*, 2nd edition, by Jean Meeus, and put the whole thing in walnut and maple.

Left
James's enclosure is inspired by the work of Braun industrial designer Dieter Rams

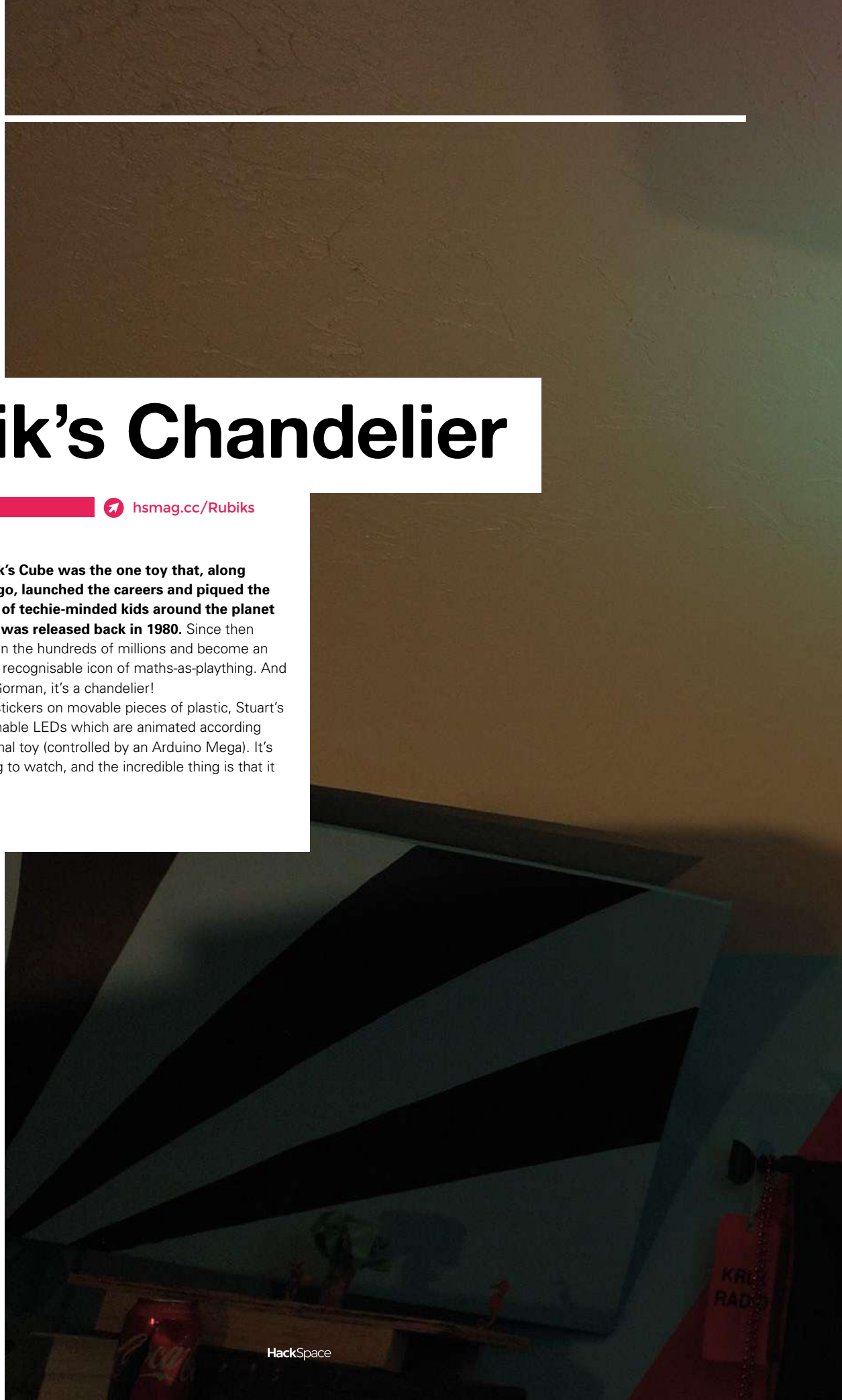
Rubik's Chandelier

By Stuart Gorman

hsmag.cc/Rubiks

The Rubik's Cube was the one toy that, along with Lego, launched the careers and piqued the interest of techie-minded kids around the planet when it was released back in 1980. Since then it's sold in the hundreds of millions and become an instantly recognisable icon of maths-as-plaything. And now, thanks to Stuart Gorman, it's a chandelier!

Instead of coloured stickers on movable pieces of plastic, Stuart's version uses programmable LEDs which are animated according to the rules of the original toy (controlled by an Arduino Mega). It's absolutely mesmerising to watch, and the incredible thing is that it solves itself.



Right
You can use Bluetooth to control the light levels, make it flash, or just have each face of the cube a solid colour

Smart Fairy Tale

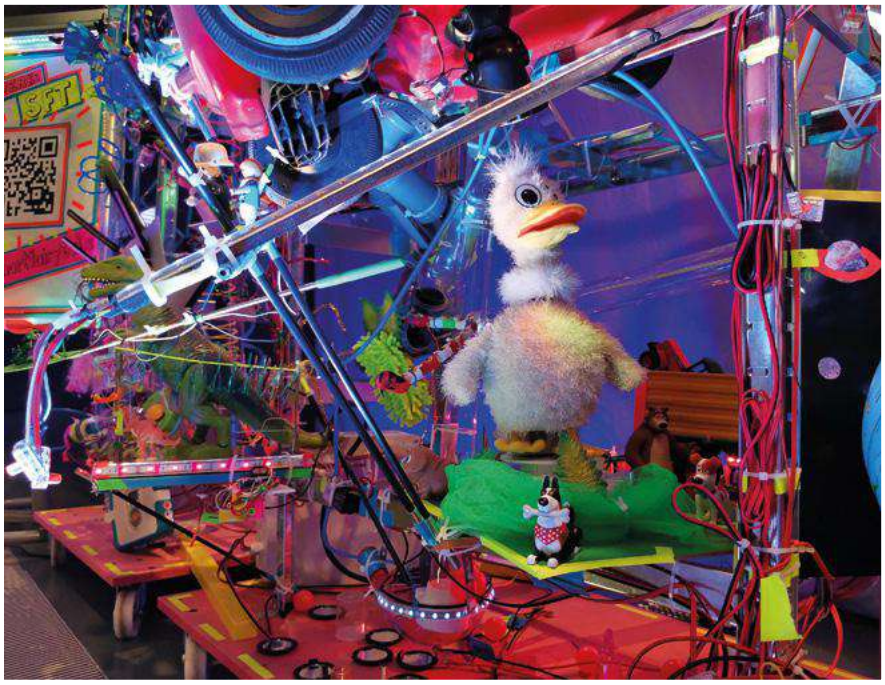
By Niklas Roy, Felix Fisgus, and many others

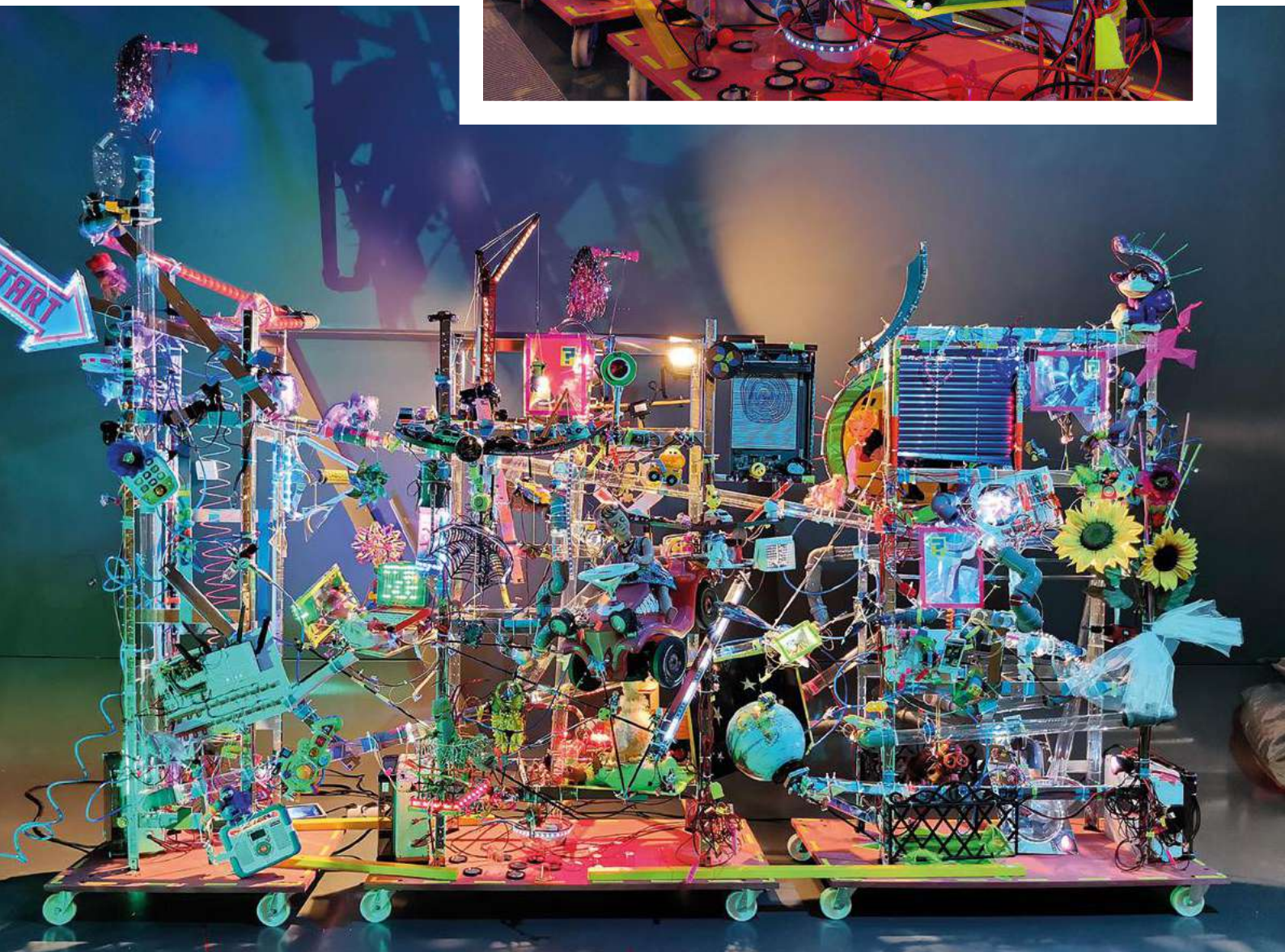
hsmag.cc/FairyTale

This installation, controlled by a combination of Raspberry Pi, Arduino, and smartphone, was created for an exhibition at the Phæno Science Center in Wolfsburg, Germany. There's a lot to it, so we'll let Niklas explain:

"'Smart Fairy Tale' is a remote-controlled story machine. A little red ball rolls through a maze of transparent pipes. Interrupting light barriers along its way, the ball triggers the movements of numerous animatronics. Different track switches inside the pipe system determine the path of the ball, leading to a variety of stories which are being told by the apparatus. Visitors can operate the installation via their smartphones."

Right
With so many moving parts, the bill of materials is complex. There's one Raspberry Pi 4 at work here, three independent power supplies, and 25 Arduino Nanos





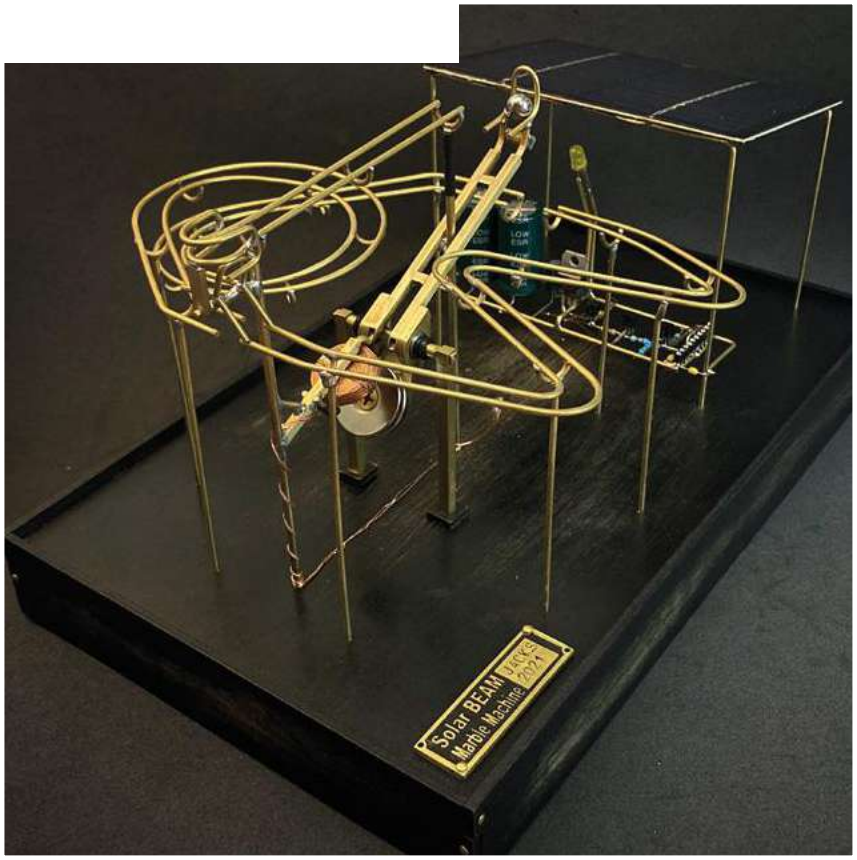
Solar BEAM Marble Machine

By Jack Spiggle

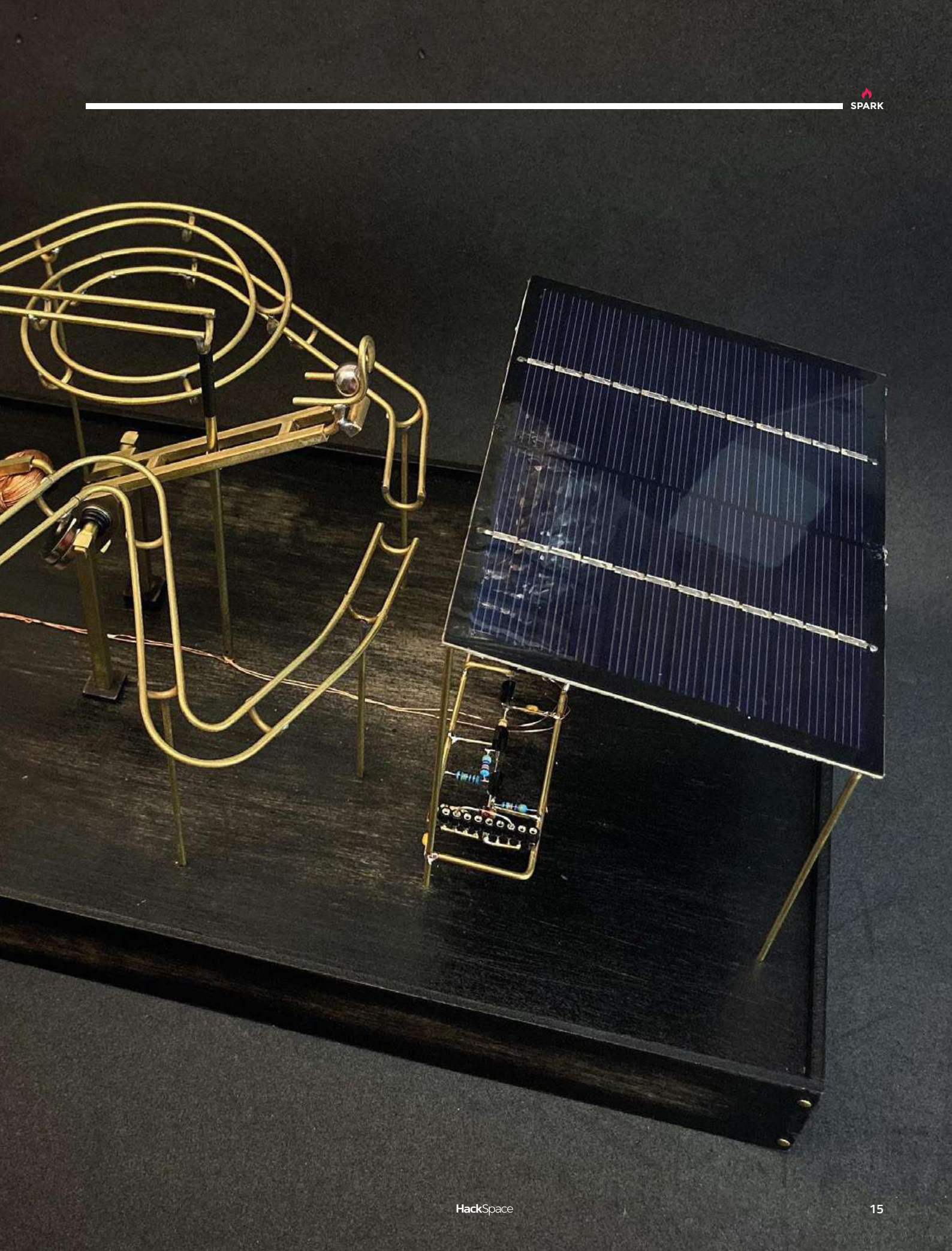
hsmag.cc/MarbleMachine

We love brass here at HackSpace towers. It's conductive, it looks great, and you can get a circuit working with it reasonably quickly, as long as you don't expect it to look perfect.

Jack Spiggle, aka NanoRobotGeek, has used brass not just in the electrical parts, but also in the mechanical parts of his Solar BEAM Marble Machine. There's a lot to it, so we'd highly recommend checking out the link but, in short, it takes in energy from the solar panel and stores it in a pair of capacitors; when the charge gets too much to store, the capacitors release it into the rest of the circuit, where it powers a solenoid that pushes a lever that moves a marble around a track. Brilliant.



Right
BEAM stands for Biology, Electronics, Aesthetics, and Mechanics, and refers to a low-component way of making robots



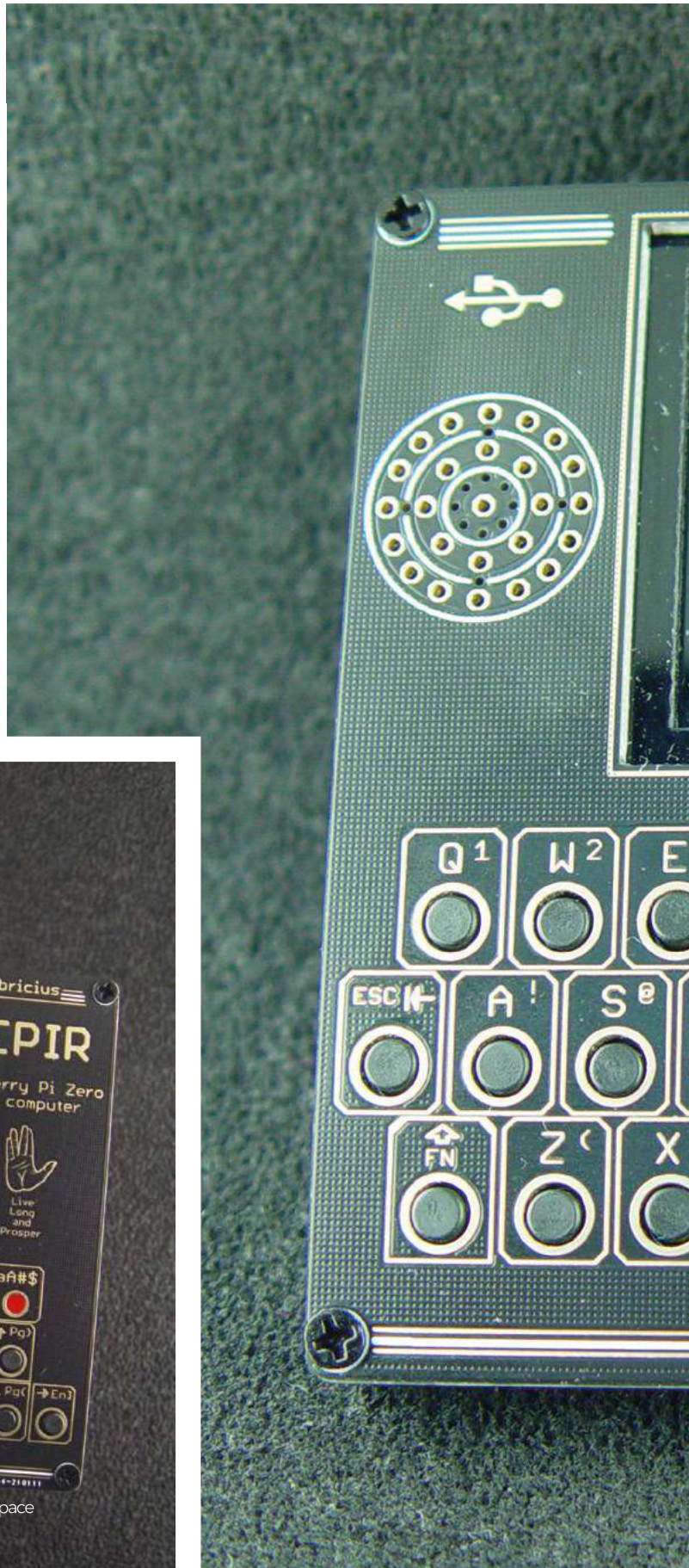
REGULAR

Zepir

By **Peter Misenko** hsmag.cc/Zepir

This is an intriguing project: it's a full QWERTY keyboard plus a screen that plugs into a Raspberry Pi to make a fully functional computer. Although its small form factor lends itself to the Raspberry Pi Zero, its creator Peter Misenko tells us that it'll work just as well on a Raspberry Pi 4.

What's great about this project (and Peter tells us that it's still a work in progress) is that it's designed to be a single-component build. The board, keys, and screen all come out of the factory in one piece. □



Below
There's no keyboard-scanning chip in Peter's device; keys are connected directly via GPIO


Bobricius
ZEPIR
Raspberry Pi Zero
mini computer
Live
Long
and
Prosper
the exact distribution terms for
each program are described in t
he
individual files in /usr/share/d
oc/*/copyright.
Debian GNU/Linux comes with ABSO
LUTELY NO WARRANTY, to the exten
t
permitted by applicable law.
Running
pi@raspberrypi ~ $
DEL
aA#$
ENT
SPC
www.PeMi.Technology - Factory For Materializing Dreams

Objet 3d’art

3D-printed artwork to bring more beauty into your life

If you have a dog, it'll tell you when it wants to eat. If you have a cat, you'd better be on guard to cater for its every whim, or it'll burn your house down, turn you into a newt, or something else unpleasant.

Sebastian Sokolowski takes looking after his cat seriously: so seriously that he designed this automatic pet feeder. It features an embedded PIR sensor to detect if the pet is nearby; a beeper to let the pet know that food is on its way; it can be controlled in person or via WiFi thanks to an ESP8266, and it's all housed on Sebastian's own custom PCB.

Sebastian has set his device up to feed his cat every day at 7am, but thanks to the PIR sensor, he can receive messages on his phone if the cat is near the feeder (indicating it's probably hungry), allowing him to feed the cat at any time regardless of what schedule is set.

And the 3D-printed part? Well, the electronics and mechanical parts are all housed in a self-contained unit that slides out of the food hopper in one part, meaning that it's incredibly easy to take apart for cleaning. □

hsmag.cc/Feeder

REGULAR

Meet The Maker: Zack Freedman - Voidstar Lab

Product designer turned YouTuber

Gentleman, scholar, explainer of complicated things, and maker of absurd, brilliant things, Zach Freedman responded to the pandemic the way many of us wanted to: he headed for the hills. Now, from his mountain lair, he uses Arduinos, Raspberry Pis, 3D printing, electronics, and anything else he feels like to build crazy things. And luckily for us, he documents his creations on YouTube (try this one if you ever wanted to know why PCBs exist **hsmag.cc/circuitsonboards**). We met under cover of darkness to find out what he's learned from ten years as a freelance product developer, and seven months as a YouTube maker.

> From his mountain lair, he uses Arduinos, Raspberry Pis, 3D printing, electronics, and **anything else he feels like to build crazy things**

"Before I started my YouTube channel, I was doing freelance product development, prototyping for about a decade. This would be about my tenth year in business if I hadn't had the career change. I cut my teeth during the second tech bubble – around about 2011–2012, there were lots of hardware startups, so work was everywhere.

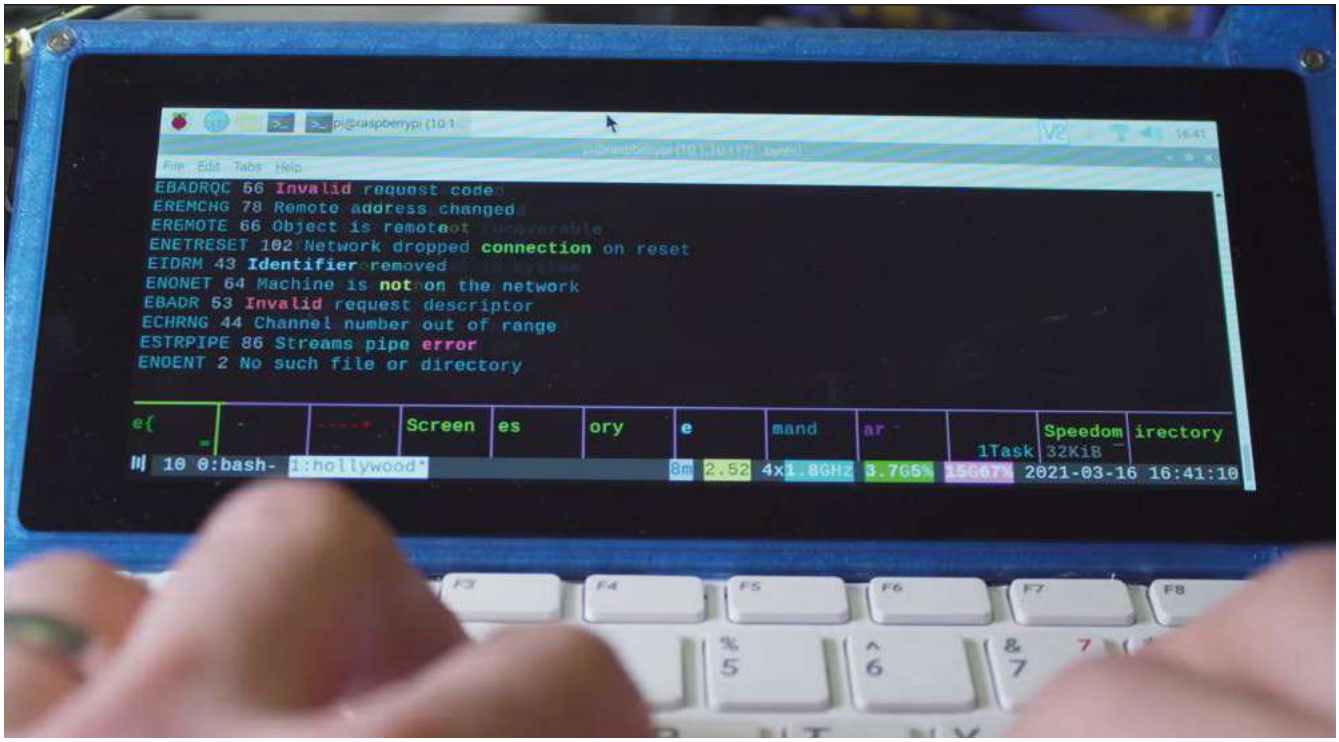
"I was learning how to make stuff, and I happened to be decent at it. I had started a hackerspace [MakerBar in Hoboken, New Jersey], lots of people were coming by trying to hire people, and I just hoovered up the jobs. Next thing I knew, I called off my job search because I had enough clients.

"I've done all kinds of crazy projects. I built a device that painted pictures using two 20 kV electron guns, held in a plotter to draw on this 12 ft × 9 ft piece of paper. That was working great until it suddenly stopped working hours before the curtain was going to go up. I solved this literally as people were standing in line waiting to get in, because I realised the humidity had dropped enough that static was affecting it.

MASS ADOPTION

"Somewhere around 1000 buildings in New York have a boiler control system that I developed… I've modified classic cars with Bluetooth radio… I've put sensors in horse saddles… people want to do the craziest things. Every time you see a light blinking on the set of *Mr. Robot*, I probably made the device that did that. If there were electrons flowing through it, I could convince people that I could do it. It was fun, and it brought us to New York City. Before, I was living in suburban New Jersey and it was just too hard to find clients. New York is where the startups are, so that's what we moved in for.

"The climate control client was a really long-running job, they kept me occupied for four or five years. That



job ended in December 2019. It had been years since I'd done a serious search for clients, so I was revving back, going to all the meet-ups, trying to get my foot in the door in the entertainment industry in New York, and right when I had a few contacts coming in and a few trade shows, speaking engagements, suddenly everything shuts down. What a time!

"Nearly every one of my clients came from a meet-up or a trade show or some other thing. I'd go to these things wearing a heads-up display or a giant smartwatch and people would come over and be like, 'hey, you build electronics, I need some electronics'.

"There was no way that could happen any more. I decided to start making YouTube videos to promote myself, to try to get some clients. The videos ended up doing well enough that I decided to go all in on it. It had to be one of the only times in my life when everything went according to plan. It takes a lot of luck.

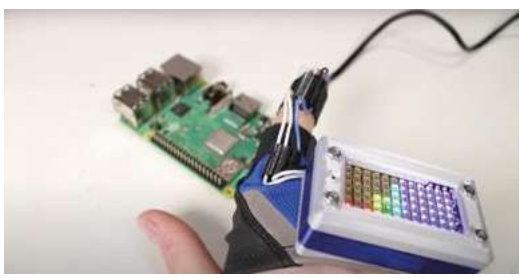
"I don't think we could have made this happen at any other time except for at the beginning of the pandemic. People are going to be stuck at home, they're going to be spending all their time online; if there was ever a time, this is the time to do it. →

Below
Zack's cyber deck uses a Raspberry Pi 400 as the basis of a cyberbunk device for on-the-move command line hacking

> 3D printers now let a schlub like me run off parts one after another **that even the most talented machinist would struggle with**

Hopefully, as life gets back to normal, people will keep watching YouTube.

"My shtick is that everything is 'practical-style'. You know how you have a kosher-style deli, and it looks like a kosher deli but they'll put Swiss cheese on your corned beef sandwich. All my projects are ostensibly useful, even if it doesn't turn out that way. Almost all my projects begin with a need for something: the cyberdeck is for hacking and interacting with stuff on the go, where you can't put down a laptop; the heads-up display is like a teleprompter, and the data glove is a way to interact with the heads-up display without having a keyboard. The Nerf blasters are because I was into Nerf wars at the time.



Right
This glove enables the wearer to feel the presence of electricity. Pretty cool, right?

"I try to include some elements of pragmatism in everything. Even the more expressive stuff, like the wall art I did and the oscilloscope, it still has some sensible use. With the oscilloscope, I put a second screen on it and all that stuff, but I try to make things useful-style. I want people to say, 'that's really cool; I can see myself using that', while at the same time not boring people with anything that's too dry and stodgy. We don't do anything just for the sake of using the technology – we always couch it in a greater need.

REALITY SHOW

"Voidstar Lab isn't fiction, but it is a hyperreality. We're not showing you how to print various filaments; we're going to print all the filaments. We're not going

to trick out a computer and add lights to it; we're going to trick out an oscilloscope.

"There's no swearing on my videos, but the original intention was not to make a family-friendly channel – I just think it's funnier to bleep it out than to say it because it leaves it to your imagination. But it turns out that that makes your channel far more approachable to sponsors and partners. Like, it makes the channel a lot easier to work with.

"The bigger maker channels, they're not always the type of things that you could show your kids. And I think it's funnier. Especially us Americans, we're so uncreative with our profanity.

"There's also no weaponry or violence on the show. We're not shocking people or inflicting pain or anything like that. And that's not for the purpose of sponsorship, it's just that making sadistic and masochistic projects is done to death. I can't compete on that level. And the no weapons thing is because it would take over the channel. There are certain people that see a gun and immediately think, 'this channel is not for me'. It goes the other way too. I occasionally get complaints when I do stuff with Nerf guns along the lines of 'why don't you be a real man and do stuff with real guns?'.

"One of the things that really fires people up about making, one of the most exciting aspects of getting immersed in someone else's projects, is the seeing this stuff that shouldn't exist come to life. Ideas too outlandish to make products out of, or too specific to build a community around. Things that seem like nobody could build them in a basement. People love seeing that sort of stuff come together.

JUST DO IT

"I hate using the phrase 'anything's possible', but anything has been possible for a very long time. This is just showing how that plays out. 3D printers now let a schlub like me run off parts one after another that even the most talented machinist would struggle with. Literally so cheaply and easily that they're disposable. Every so often, I'll stop what I'm doing and print whatever I've got on the screen, knowing that I'm going to throw it out, just to →

Above
With dimensionally accurate 3D printing, you too can build a Nerf sniper rifle

make sure that I've got my dimensions right. It's just so accessible, easy, and fast.

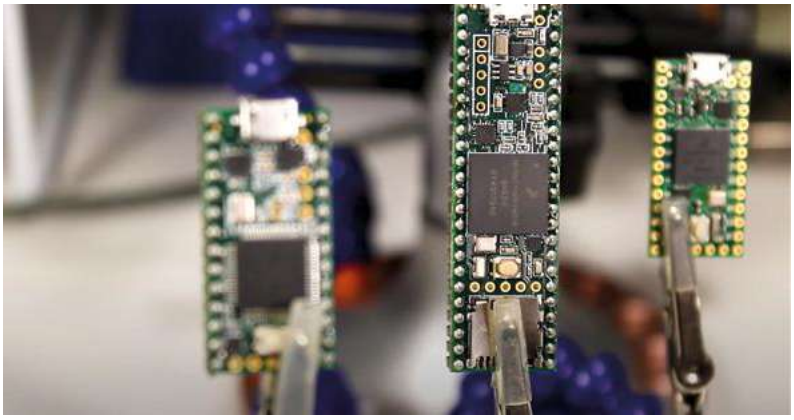
"More than that, it's also that the information is not just more easily available but also explained better. Linux, right, the way that line works and what you can do with it is basically unchanged for the last 20 years or so. But nowadays, you can get tutorials and code snippets instead of having to figure it out all on your own. For me, it makes or breaks whether I could use it or not. I tried to use microcontrollers before Arduino got decent, and if you are not used to embedded development, it is a nightmare to even set up the toolchain. If you already have code, getting it onto the microcontroller is diabolically difficult, or at least it was back in the day. Now you

> "It's incredible – technology has never **been cheaper, more accessible, or better explained**"

Right
Before discovering Zack's channel we had no idea of the link between WW2 and the prevalence of printed circuit boards. And yes, that is a home-made Google Glass-like device he's wearing

just plug into your USB port, you press upload, and that's it. In fact, for some of these, you don't even need the software; you can do the whole thing online. The tutorial and the IDE are on the same web page. It's incredible – technology has never been cheaper, more accessible, or better explained.

"The maker community, both at the individual viewer level and also at the content creator level, has been infinitely more friendly and welcoming than I expected. The table is clearly big enough for a lot more people to eat at. There's always more room for stuff.



WE'RE ALL MAKERS

"I think most people are makers in the context of something else. Mechanical keyboards, or custom guitar pedals, or Nerf blasters, and if you were to say to them, 'you're a maker', they'd probably say 'I guess so'. I think the maker community is a lot bigger than the maker community thinks it is, and there's a lot more interest in a lot of different things. That's the way to get people into it. It's not 'hey, let me teach you how to solder', or 'let me teach you how to 3D print'. It's offering to show someone how to build the most crisp-feeling mechanical keyboard that you have ever felt. Or like, 'hey, do you want to know how to break into your router like a Hollywood hacker and snoop on people's security cameras?'. That's a much better draw than, 'hey, do you want to learn to use the command line?'.

"That's what makes things like Raspberry Pi so powerful and able to bring so many people in. There's a huge difference between saying to someone, 'hey, you should try Linux; it's open-source!'. And saying, 'hey, you want to make that arcade cabinet with all those games? Here's this Linux device that makes it possible'. And now that it's easy enough that it's actually practical, it's great. So many people are getting into making stuff and they don't even realise it.

"I hope these hobbies that people have picked up in the various lockdowns stay with them after things spin back to normality. This last year has been a lot of people's first taste of making things themselves, and the first time they've been able to experiment without any possibility of someone looking over their shoulder and judging them." ◻

SFERALABS

Raspberry Pi goes industrial

Long term availability, high reliability and support

Tailor-made embedded solutions

Our range of Raspberry Pi-based devices is wide. Really wide.
But what if your project needs something more?
We can put all of our expertise into your custom OEM design.
Don't hesitate to contact us to explore every possibility!

Visit our website www.sferalabs.cc

POWERED BY Raspberry Pi

Open Hardware Summit:

from Covid PPE to open chip designs

Meeting makers in a virtual world

Drew Fustini

@pdp7

Drew Fustini is a hardware designer and embedded Linux developer. He is the Vice President of the Open Source Hardware Association, and a board member of the BeagleBoard.org Foundation. Drew designs circuit boards for OSH Park, a PCB manufacturing service, and maintains the Adafruit BeagleBone Python library.

One of the few silver linings of this pandemic has been how easy it has been to drop into cool events and talks all over the world without needing to get out your passport. One of the highlights for me recently was the Open Hardware Summit, which had a range of excellent talks from all over the world. Some of my personal favourites were Oluwatobi Oyinlola's talk about the Nigerian open hardware community, Julieta Arancio's talk about open science hardware in Latin America and LeeLee James's highly entertaining Q&A about her Twirling Tech Goddess YouTube channel (seriously, go subscribe!).

The keynote speaker was Meghan McCarthy, who works for the U.S. National Institutes of Health (NIH). It was fascinating to hear about her work running the NIH 3D Print Exchange, a resource for sharing and creating 3D-printable models related to bioscience and medicine, including open hardware for use in laboratories. The stories she shared about the swift, collaborative response to the pandemic from her team, and the medical, manufacturing, and maker communities were inspiring.

If you've been following my recent columns about open chip design, you might be interested to watch the open hardware chip design panel I moderated as part of the Open Hardware Summit. The panellists (Megan Wachs from SiFive, Mohamed Kassem from efabless, iconic hardware hacker Bunnie Huang, and Jason Kridner from the BeagleBoard.org Foundation) brought an incredible range of experience to the table. The panel discussed how Migen and LiteX provide powerful abstractions that help people from a software background design FPGA projects, and considered what practical improvements can be made to help people work collaboratively on open-source chip designs. Mohamed also shared some exciting details about the first 40 open-source chips to be made on the free Google SkyWater shuttle.

Open Hardware Summit has announced plans to be in New York City in 2022. I hope that they are third time lucky!

This year's talks and panels are all online at **hsmag.cc/2021OHS**, and you can find information about the speakers at **2021.oshwa.org**. After the cancellation of the 2020 event and the online-only event in 2021, the Open Hardware Summit has announced plans to be in New York City in 2022. I hope that they are third time lucky!

Letters

ATTENTION ALL MAKERS!

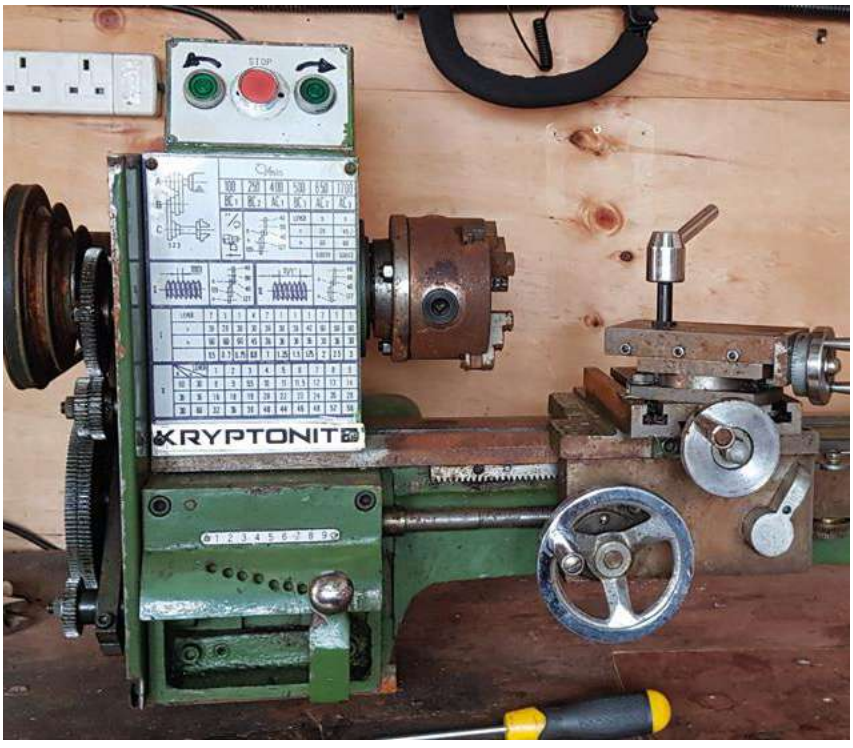
If you have something you'd like to get off your chest (or even throw a word of praise in our direction) let us know at **hsmag.cc/hello**

REPAIR REVOLUTION

I read with interest your cover feature (and Andrew Lewis's tutorial) on repairing old machinery. And it was surely no coincidence that the EU has announced 'right to repair' laws coming into force from 1 March, and the UK government following them this summer. It's almost as if fixing things, rather than throwing them away, makes sense.

David Rogers
Whitstable

Ben says: From an ethical, environmental, and financial viewpoint, repair wins hands down over buying new. But the aspect of repair that really strikes a chord with me is it's also about respect for the craftsperson who made the object in the first place. Everything made by hand carries a bit of the maker with it, so repair is also an act of love.



FIRST ATTEMPTS

Are you seriously trying to tell me that the first-ever 3D-printed project by Estefannie was a helmet with a build time of 136 hours? What's wrong with a wobbly Benchy, or a stand for holding wooden spoons, or something like the rest of us struggle along with? Why are people so good at things I find so hard? See also: Steph Piper's ludicrous claims that she'd never designed a PCB before producing her works of art. My first PCB was a bit of copper that I drew on, then put in some sort of solution to dissolve everything but the traces. It was a long time ago, so I can't remember all the details, but I know it wasn't anywhere near as good as her stuff. I take my hat off to the pair of them, and to anyone else learning new things.

John Shipley
London

Ben says: Zack Freedman, who we spoke to for this month's Meet the Maker, puts it like this: "technology has never been cheaper, more accessible, or better explained". Anyone learning now has a massive advantage over anyone learning 30, 20, or even 10 years ago, and that's exactly as things should be if we're ever going to get the jet packs we were promised.

WHOOPS

Re: HS42 email "Also inside: The prettiest PCBs ever committed to silicon". PCBs aren't made out of silicon. Not even Steph Piper's.

Chris W.

Ben says. You're right. But as the great man sang, two out of three ain't bad: they are pretty, and they are PCBs. If only we knew an expert in PCBs who could have put us straight on this?

CROWDFUNDING NOW

Choc mate 2

Automate your decorations

From €4800 | **kickstarter.com** | **Delivery:** October 2021

Chocolate has one key similarity with 3D printer filament: it's solid at room temperature but liquid a small amount above this. Because of this, people have been trying to 3D-print with chocolate almost as long as there's been a hobbyist 3D printing community. It works, to a certain degree, but the devil is always in the detail. The way a substance flows, cools, and solidifies is just as important as the temperature it melts at – and getting quality chocolate into an intricate shape has been difficult.

Germany-based chocolate^3 claims to have cracked this particular problem with the choc mate 2. At 4800 euros, this solution comes with a hefty price tag (there isn't an option to buy just the extruder to upgrade an existing 3D printer).

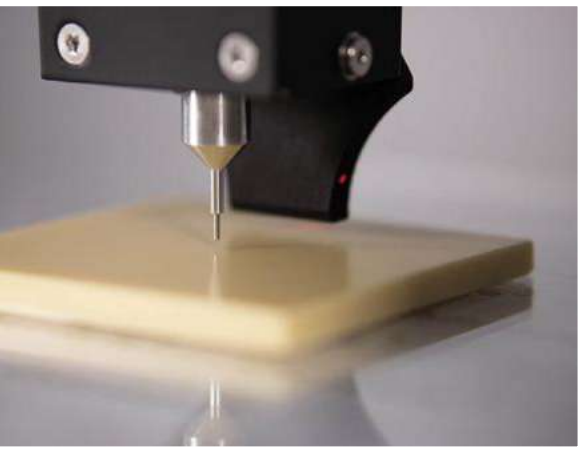
If you're spending this sort of money, there has to be a real use for it beyond simple novelty. This is a real struggle with food 3D printers. As overhangs are tricky (there are no significant overhangs on any of their examples), building complex structures is almost impossible. Print speed is another barrier to larger creations. By far the largest use of these 3D printers seems to be to create expensive novelties for corporate events. However, that may just be because the right creative use for them hasn't been found yet.

At the time of writing, the project has raised 50,000 euros. This is past their initial goal, but if they storm ahead to 250,000 euros, they will offer a dual-head printer as an optional upgrade. This could give the ability to create some really interesting patterns in a combination of different coloured chocolates. Depending on how well this works, and how long it takes, this could open up significant opportunities for creative chocolatiers.

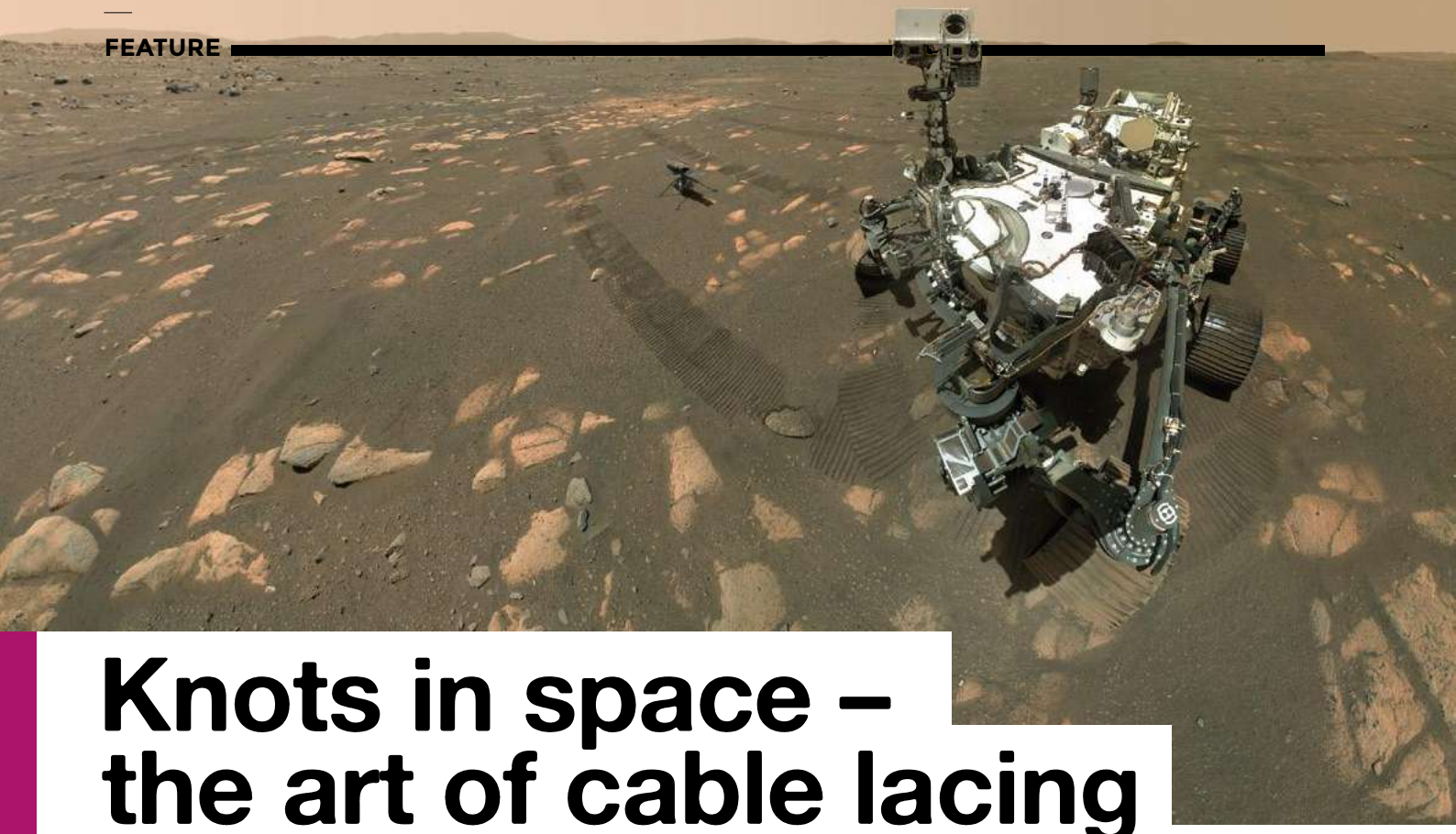
It's great to see innovation in 3D printing, and this isn't just about chocolate. The techniques needed to make this food work may end up being useful for a whole host of other substances that may (or may not) be more useful. Perhaps we're a bit cynical about the use-cases for chocolate 3D printers. If you have a budding idea for a great way to use this, you will soon have a new option to choose from to help you build your sweet creations.

BUYER BEWARE

When backing a crowdfunding campaign, you are not purchasing a finished product, but supporting a project working on something new. There is a very real chance that the product will never ship and you'll lose your money. It's a great way to support projects you like and get some cheap hardware in the process, but if you use it purely as a chance to snag cheap stuff, you may find that you get burned.



FEATURE



Knots in space – the art of cable lacing

A zero-maintenance approach to wire routing

Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

Figure 1 The Perseverance rover selfie, with the Ingenuity helicopter ready for its test flight

Credit NASA/ JPL-Caltech/MSSS

On 6 April 2021 we, at HackSpace magazine headquarters, were thrilled to see this amazing selfie (Figure 1) that the Perseverance rover took of itself, and the phenomenal Ingenuity helicopter project it had jettisoned to try to perform the first-ever powered flight on Mars. It's an amazing image full of inspiration, but our eyes are always drawn to a certain feature! Zooming in on the image, (**Figure 2**), we get some really nice detail of cables on the Perseverance rover, and you can tell that the periodic bindings aren't cable ties: they are in fact knots tied in a special tape. We're going to explore these space-grade cable lacing techniques.

Cable lacing is an old art, used since the early days of telecoms, aerospace, and other sectors. It only became diminished as a craft after 1958, with the invention of cable or 'zip' ties. However, there are still use cases for cable lacing and, indeed, it offers some advantages over the near-ubiquitous cable tie. We are going to look at the NASA standards for cable lacing, and the knots and lacing techniques, but it's true to say that lots of other sectors, such as maritime and telecoms, have their own variants on cable lacing if you want to explore it further.

WHY CABLE LACING?

Lacing offers numerous advantages over cable ties. One of the most common cable lacing techniques is spot tying, which replicates a cable tie in that it holds a bundle together at a single specific point. Spot tying offers a few significant advantages over a cable tie. One is that cable tie design is such that it can create an uneven force on a bundle of cables and, over time, can dig into cable insulation. Another factor can be weight, in that cable ties, particularly high-quality ties, can be heavier than a small piece of cable lacing. Other reasons are that cable lacing can be undone and redone with relative ease and, finally, often cable ties increase the diameter of the cable bundle at the tie point. This last point may be relevant if the cable loom has to be installed via small apertures etc. It's also worth noting that cable lacing can be undertaken with a variety of materials – objects travelling in space may need to be tolerant of huge temperature differences, exposure to radiation, and also not outgas any waste particles when in the vacuum of space.

We've been lucky enough to track down some different types of cable lacing, including some aerospace-grade Nomex Lacing Tape which is very tough and can survive extremes of temperature well. We also have some thin fibreglass tape which is used

a lot in aircraft and also in high-performance engines in racing cars etc. The natural Nomex tape we have is specified for use over a wide range of temperatures and is impregnated with liquid nylon. The other fibre or 'E-Glass' lacing tape is PTFE-coated, presumably to stop any wearing of the cable harness, and it's also specified in the data sheet that it will never elongate more than 5% of a given length. The Nomex Lacing Tape we have also shows how strict aerospace and space engineering quality control is. Each batch of the tape has a unique batch code and, on request, you can get data from stringent tests carried out on a sample from your manufacturing batch, rather than just test information on the general product (**Figure 3**).

As the lacing tapes are quite fine, they produce a very small knot or tie. We've shown some examples using the lacing tape in some images. We've also used some paracord around a short piece of dowel to tie the example knots for visibility.

The classic cable lacing technique is to use a spot tie (**Figure 4**). Spot ties are applied individually along the length of the cable bundle in a similar way that one might apply cable ties. NASA-compliant spot ties are formed with the creation of two knots on top of each other, a clove hitch, and a square knot. The basis of the spot tie is a clove hitch. The clove hitch is a fantastic knot for cable tying applications as its two turns around the bundle are tensioned equally with the opposing ends, creating an even pressure on the cables. It also

Figure 2
Zooming in on the Perseverance rover, we can see the cable lacing techniques being used to manage the cable harnesses

Credit NASA/ JPL-Caltech/MSSS

is a type of knot that can be tied around an object even if you don't have an open end of the object. For example, if you had a bunch of cables that created a sealed circle, or a cable loom already installed, you could still apply a clove hitch at any point. To tie the clove hitch, keep a small length in hand. We'll call this end of the cord our 'tie-off end', and then wrap the other 'running end' around your cable bundle or practice item.

As you bring the cord around, cross it over the tie-off end and continue to make a second turn. As your second turn comes around, take the end in hand and thread it under the second loop (**Figure 6**). Cinch this clove hitch up tight, moving the loops together if needed.

The completed clove hitch is finished into a NASA-compliant spot tie by the addition of a square knot tied on top of the clove hitch. Outside of NASA cable lacing, this knot varies from a common granny knot/reef knot, as the intention of the knot is to lock the tension of the clove hitch, but for cable lacing →

YOU'LL NEED

- **Some cables or a piece of broom handle**
- **Some cord or thin ribbon**
- **Pair of scissors**

Figure 3
Professional lacing tapes come in a variety of materials tailored for different uses and different working environments

ONLINE RESOURCES

There are lots of resources and interest in cable lacing online. The NASA standards for cable management, which include the approved cable lacing techniques, are documented in a NASA Technical Standard document 8739 titled '*Workmanship Standard for Crimping, Interconnecting Cables, Harnesses, and Wiring*'. It's available as a free PDF download for public use and is an interesting read – not just on lacing, but on connecting and splicing cables, using braided sleeves, and more. This standard is where we have taken the sketches from depicting the knots that we've explored for cable lacing.

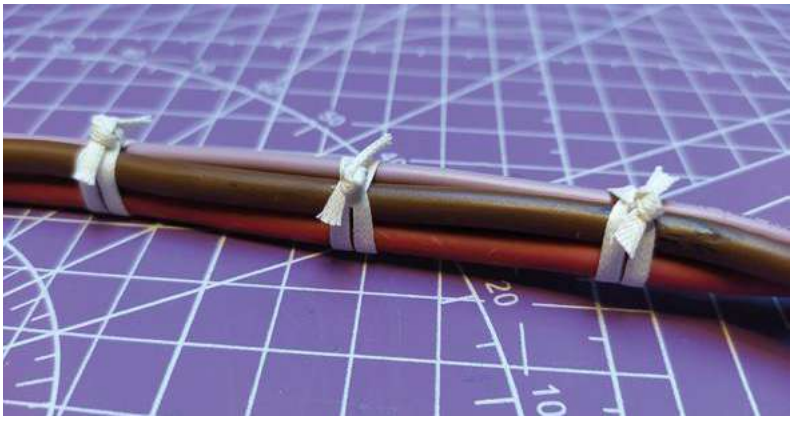

NASA TECHNICAL STANDARD

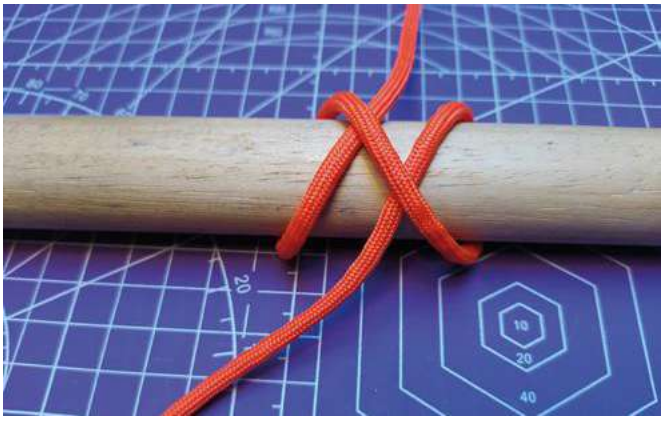
NASA-STD 8739.4A with Change 2

National Aeronautics and Space Administration Washington, DC 20546

Approved: 2016-06-30 Superseding NASA-STD-8739.4 With Change 6

WORKMANSHIP STANDARD FOR CRIMPING, INTERCONNECTING CABLES, HARNESSES, AND WIRING





> **The added bonus is that every time we lace a cable,** we think of space exploration

Figure 4
Using some scrap cables, we spot-tied them using the NASA-compliant spot tie knot in the E-Glass/PTFE lacing tape

Figure 6
The clove hitch section of the spot tie ready to be cinched tight

Figure 9
Three knots that create a single strand running lockstitch.

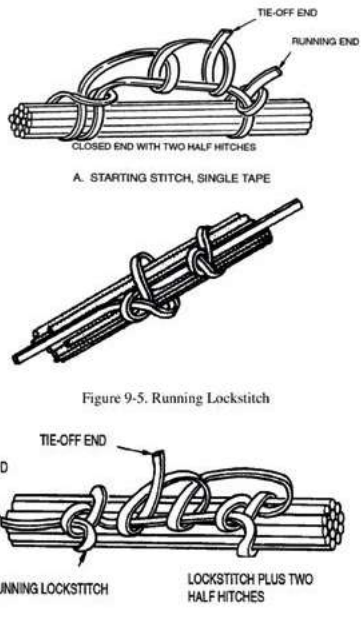
Credit, NASA

aficionados, which of these is best, is a debate that runs deep! The square knot is formed by taking the running end of our cord coming from the clove hitch, and bringing it over to form a loop, and then our tie-off end goes under the running end, and then over and through the loop we formed (**Figure 7**).

Cinch that half of the square knot tight, and then bring the tie-off end over to form a loop, and bring the running end over the top of the tie-off end, and then up through the tie-off end loop (**Figure 8**). Cinch this tight.

Another common application of cable lacing is using a 'running lock stitch'. A running lock stitch is where the same single or double strand of lacing tape continues along the cable bundle, creating a kind of running stitch. We've only looked at the single-strand stitching, but double-strand stitching is of use if you have junctions in cable bundles, as you can continue down each branch, changing from a double tape stitch to a single tape stitch.

One benefit of a running lock stitch is that it reduces the number of operations needed by a technician who is lacing a larger project. The downside is that you can lose tension in the stitch system until it is fully tied. If you let go after twelve stitches, you may need to start again! The simpler



SPACE SPACING

Spacing between spot ties, running lock stitches, or cable ties is specified in the NASA standards, and is linked to the diameter of the cable bundle. For example, for a cable bundle of 6.4 mm or less, the stitches, however formed, should be no more than 19.1 mm apart. Moving to larger diameters and the spacing increases. For example, a 25.4 mm bundle spacing will be a maximum of 50.8 mm. For any cable bundle diameter over 25.4 mm, the maximum spacing has an upper limit of 76.2 mm.

single tape running lock stitch has three components – the starting stitch, a running lock stitch that can be tied repeatedly, and a final closing stitch of which there are a couple of varieties in the NASA document. There is a specific start stitch suggested, shown in **Figure 9**. Note that all of the diagram marked A is the starting stitch. The part on the right-hand side looks like it is the first running stitch, but actually, it is a slightly different knot than the subsequent running stitches. Our approach to tie the starting stitch was to start with a generous part in hand that would become the tie-off end, and start with the part where that end

Figure 7
Forming the first half of the square knot

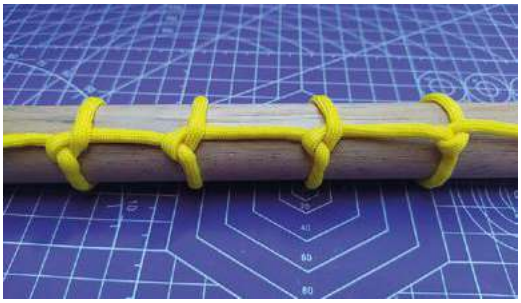
Figure 8
Finishing the second half of the square knot

in hand passes under the cable bundle. We found we could follow it through using the diagram, tying the right-hand side adapted stitch in the image, and then finally adding the two half hitches (**Figure 10**).

When dealing with running varieties of cable lacing stitches, we have a 'running end' which is the end of the line that is actively running to the next stitch, and a 'tie-off end' which is the end on the non-working side of the starting and finishing knots. The NASA standard also indicates in the text that our spot tie (the clove hitch with a square knot on top) is an acceptable starter stitch for a running system, with either end of the square knot as the running end. So, feel free to use that – your space vehicle can still fly with it!

The running lock stitch is fairly straightforward but also easy to tie incorrectly. In **Figure 11**, the first three running lock stitches from the left are tied correctly, but the last one on the right is incorrect. It has the common mistake of being tied with the running end travelling over the top of the loop towards the next stitch. If you tie a few of these incorrectly in a row, you'll notice that the running section between the stitches is floating above the cable bundle rather than

Figure 10
Following the diagram, we carefully tied the starter stitch



in contact with it, but it's harder to spot if you make this mistake with the very fine cable lacing tapes. Finally, we come to the closing stitch. The closing stitch is pretty straightforward – simply tie two half hitches into the running section before the final lock stitch and cinch everything down tight (**Figure 12**).

We hope that you've enjoyed this exploration of this fascinating art. Whilst we still might occasionally use a cable tie, we imagine many of our projects will end up with cable lacing on wiring harnesses after learning these techniques. It's fun, quite quick to do with a bit of practice, it's re-doable should you need to make a change and, of course, reduces usage of one-time-use, plastic cable ties. The added bonus is that every time we lace a cable, we think of space exploration. ◻

SPACE-RATED EPOXY

You might see, in the NASA standards PDF, that there is the occasional mention of 'staking'. Staking in engineering or machining is where an interface is made permanent by the flaring of an undersized stake in a hole to make it an extremely tight fit. In terms of cable lacing, staking similarly means to make the knot permanent and it is achieved by the addition of a staking compound – an adhesive, to make the knot permanent. For home cable lacing, a spot of superglue or perhaps a blob of a rubber-based adhesive would suffice. In space applications, however, only certain substances may be used. This is often due to the issue of outgassing. Outgassing is what happens when you place an object in a vacuum such as in space. Many materials will release molecules of matter in a vacuum, including glues, even when cured. This can be incredibly dangerous and also incredibly costly. If your staking compound leaks a film of sticky molecules that then attach to some million-dollar sensor, you might be in trouble! NASA and other space agencies have approved lists of materials that can be used in different applications and a common staking compound for cable lacing is a low outgassing epoxy resin such as 3M Scotch-Weld Epoxy Adhesive EC-2216. Be warned, though: as this is space-rated epoxy, it is priced accordingly!

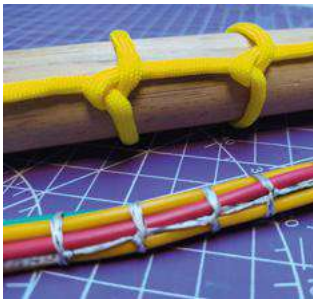


Figure 11
An example of the running lock stitch with the last one on the right tied incorrectly with a common error

Figure 12
Our example running stitch in paracord, and a bunch of cables stitched with the same knots using the Nomex Lacing Tape

Build a Makerspace for Young People

Join our **free online training course on makerspace design** to get expert advice for setting up a makerspace in your school or community.

Sign up today: rpf.io/makerspace

HackSpace
TECHNOLOGY IN YOUR HANDS

LENS

HACK | MAKE | BUILD | CREATE

Uncover the technology that's powering the future

MAKE THE INTERNET OF THINGS WORK FOR YOU

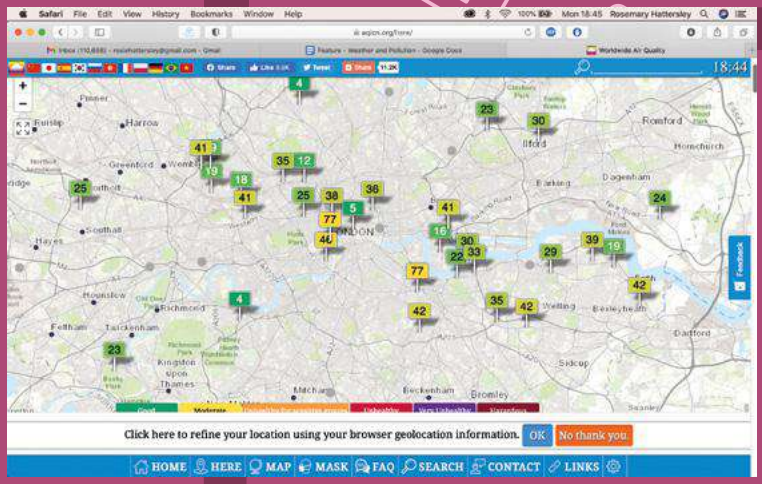
Stop trading conveniency for privacy. **Rosie Hattersley** explores 10 ways to take charge of your own smart tech

Standing in my kitchen making yet another lockdown loaf of sourdough, I've one eye on my Alexa-enabled smart clock when I get a notification from the Dark Sky app on my Apple Watch that the contents of that foreboding cloud will be arriving overhead in two minutes. Just as I dash back in with my washing, the ring on my Alexa speaker turns orange and announces that the Amazon delivery scheduled for today has just been delivered to my doorstep.

I hadn't even had to open my purse to produce my bank details – almost as soon as I'd placed the item in question into the online basket, a third-party app had pinged a numerical code to my smartphone so I could complete the transaction. Like so many recent purchases, this one was prompted by some timely advertising on social media, followed by an email reminder a few hours later about the item in which I'd shown an interest.

Google, Amazon, Facebook, Apple, and Microsoft know so much about each of us already, it barely seems like another step to give them access to our photos, location and movements, and contacts when we want up-to-date information and one-click shopping. Clawing back control of my online privacy is going to mean rethinking – and potentially rebuilding – my personal Internet of Things. Here are ten ways to get smarter with smart tech. →

KEEP TABS ON YOUR ENVIRONMENT



If you live in an urban or industrialised area, you may well be concerned about pollution levels and air quality in your neighbourhood, but less keen on constantly reminding Google or Apple Maps where you live. You can check on current particulate levels via the Air Quality Index (**aqicn.org/map/world**). Click the Here tab to find your nearest tracker, and click 'No thank you' to the option for more precise geo-located details. Clicking Accept stores your info for just one day.

If you have coding skills, you could potentially use details from the Air Quality Index to pull into a magic mirror or other display. Alternatively, you could set up your own air monitor using the detailed instructions by our very own Ben Everard: **hsmag.cc/AirMonitor**.

Here, a Raspberry Pi of any stripe, plus an SDS011 particulate sensor, work in tandem with DustViewerSharp software to explain the significance of the results. A plastic tub is used to protect the sensor, and there's the option to save data to the Raspberry Pi's SD card rather than use a free online data store such as Adafruit IO, if you prefer to keep everything stored locally.

The monitoring device will be mains-powered, so you'll need to cut a hole in the tub to run the USB power cable. Siting the contraption is also critical: mount it four feet or more above the ground, and away from any obvious source of pollution such as a flue or tumble dryer extractor.

If you prefer to build an Arduino-based air monitor, Yves Morele's particles detector project **(hsmag.cc/YvesMorele)** uses NeoPixels to indicate the status of the current air quality.

To ensure your pollution checker is accurate, calibrate it against the readings of your nearest monitoring station – usually the nearest city or large town.

Once it's set up, you can optionally add an entry for your air pollution monitor to the AQI register so others can benefit from citizen-sourced data.

Above
You don't have to tell the Air Quality Index your exact location in order to get useful information about pollution levels

Right
Arduino and NeoPixels provide a visual indication of the current levels of air pollution

SMART HEATING

Having a smart temperature controller in your home is a great idea, as you can see what you're spending, set heating levels based on whether it's a weekend or week day, the time of year and so on. Installation can be pricey, though, and some setups are definitely more user-friendly than others. Unfortunately, data leaks have led to security concerns around Nest and other well-known smart heating controllers, although the information such devices access is what enables customers to take advantage of lower running costs. If you have such a device, try disabling the microphone (and thus the Alexa functionality), and be prepared to get up and turn the dial if you want to give your front room an evening boost of warmth.

If you prefer to take things into your own hands in terms of cost-cutting, there are lots of useful widgets on the IFTTT site that can remind to start your washing-machine, dishwasher, and tumble drier to operate when electricity tariffs are at their lowest, potentially saving you a tidy sum. Just search the site for 'home heating', 'washing cycle', or other terms – **hsmag.cc/IFTTTRemind**.

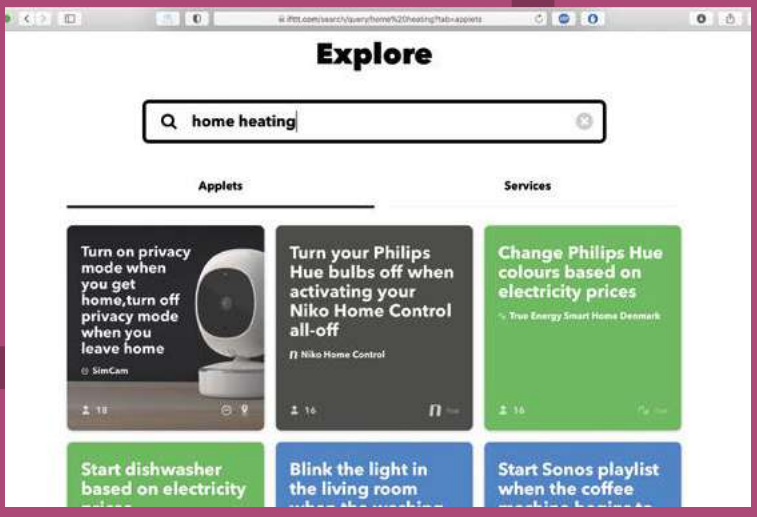
For a simpler smart thermostat option, try NotEnoughTech's cheap fix, which works for almost any standard household thermostat. Matt took an old three-wire Honeywell thermostat and added a SONOFF home automation device as a temperature sensor. Although controlled via a smartphone app (with optional voice control via Alexa or Google Home), the £5 IFTTT-enabled

addition to his home heating setup provides at-a-touch temperature control, as well as informing him of the current temperature.

NotEnoughTech on Hackster provides a useful setup guide for the SONOFF smart switch and DS11 temperature sensor – **hsmag.cc/thermostat**. RF can be used as an alternative should you prefer to use a remote control rather than a smartphone to make heating adjustments. →

Left
If you're concerned about snoops getting your personal data, try a simple thermostat hack using a SONOFF sensor and smart switch

Below
The IFTTT website offers a searchable set of applets that can be used to set appliances to run at cheaper times of the day



GET UP-TO-THE-MINUTE MAGIC MIRROR UPDATES

A firm favourite and a great choice of project, a magic mirror informs you of lots of the time, travel, news, weather, and calendar notifications that you might get from consulting Alexa or looking at your phone or Google. Created and maintained by enthusiasts, with a magic mirror you not only get away from staring at a tiny smartphone or tablet screen, you can specify your own data feeds.

Information appears to float above the mirrored screen, hence the name. Once you've read the rest of this article, you might like to add some of the monitoring and informative feeds we mention.

To set up a magic mirror, you will need a suitable screen (an unwanted laptop screen or an unwanted tablet), some observation mirror acrylic (which is semi-reflective), and a picture frame to fit it, or wood to make one, or a 3D-printed frame. You'll also need either an Arduino or Raspberry Pi with on-board wireless or an add-on WiFi module. Create a sandwich of the observation screen and tablet/laptop screen, then secure it in place inside the picture frame with gaffer tape. Note that larger screens will need securing in place with mounting points. Depending on the depth of your picture frame, you may need to add depth so that there's room to accommodate your Arduino or Raspberry Pi, and will also need to consider where to place the USB power cable so it's not seen when the mirror is hung up and in use.

Helpfully, the code you need for Magic Mirror is downloadable from the open-source project site **magicmirror.builders**, along with almost any module you care to mention. This means you can customise your mirror with the feeds you really want to see, as well as getting ideas from other builders. Ideas for frames to match your décor, how to disguise them as a TV. See the piece we wrote about an attempt to add a magic mirror-type overlay to a standard TV display so you don't need a dedicated screen for it, or even as an LCD chalkboard (**magpi.cc/105**), mean this useful setup can be as unobtrusive as you like.

Depending on how serious you are about privacy and avoiding giving your details to big tech sites, you can even add home automation via Alexa commands and face recognition: **hsmag.cc/EbenKouao**.

The beauty of a magic mirror is that you can customise it to display whatever is important to you. Local weather? Transport details? The current position of Plymouth Argyle in the league? With a DIY build, you're in control.

Right
SmartBuilds.io's Eben Kouao offers a handy demo on adding face ID to your magic mirror

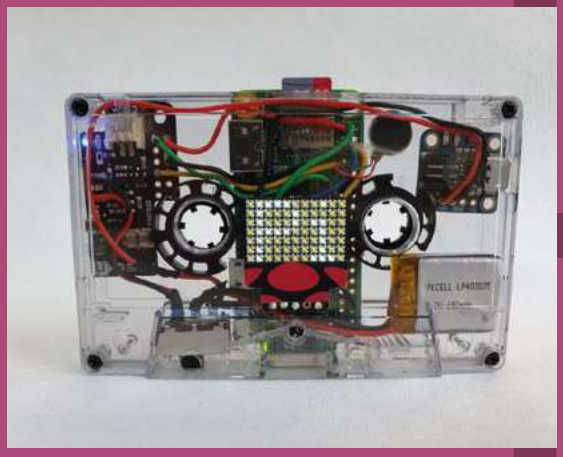
Below
Maker Dede Mitchell decided to embed her magic mirror in a chalkboard so she could check her schedule while grabbing a coffee

NEAT NOTIFICATIONS

If a magic mirror is too in your face and too much screen time is the bane of your life, Martin Mander's Cassette Pi IoT Scroller offers a whimsical alternative to notifications (hsmag.cc/cassette).

Housed in a 1980s cassette tape player, the project uses IFTTT and Adafruit IO along with a Raspberry Pi Zero and some Python code. When you receive a notification, an ominous rattle will draw your attention to the LED matrix displaying the latest not-to-be-missed social media status update or weather warning. Martin chose the Pimoroni 11×7 LED matrix because it was conveniently the same size as the window part of the cassette tape. A LiPo SHIM and battery take care of powering the scroller. Thanks to his bespoke Cassette Pi Python code, Adafruit IO receives Martin's chosen feed of updates every few seconds. These are added as applets selected on the IFTTT site, deftly avoiding settings and notifications menus on a smartphone.

Although a rather niche build, it's a great demonstration of how to stay abreast of things without defaulting to Google, Apple, et al. →



Left
A mini LED matrix and Raspberry Pi Zero proved the ideal size for a cassette-based notification device

Left
At first glance, the cassette tape player looks like a rather scruffy old music machine, but is actually primed with IFTTT applets and linked to Adafruit IO

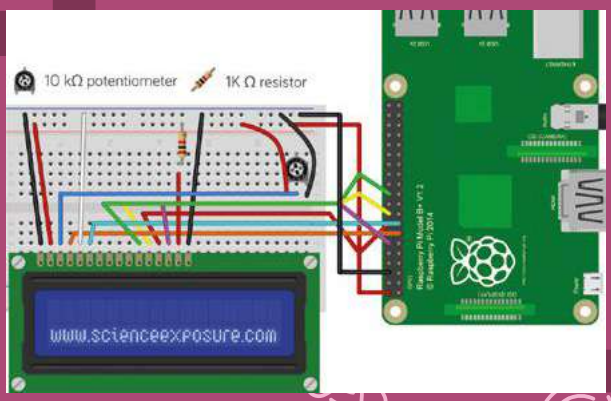
PUBLIC TRANSPORT TRACKER

Right
With TfL buses all using GPS trackers, making use of real-time tracking code means you can ignore your smartphone and build your own arrivals board

Below
The countdown API for trains and buses can be accessed and used in your own live departure board project

Different towns and cities seem to use different systems for providing up-to-date information on bus and train services. With some, you need to text a request from the bus stop where you're waiting, while others offer a digital countdown that's more or less accurate. How much more convenient, though, to be able to view such detailed information from the comfort of home before setting out or, as per one cherished local pub, by glancing at the digital display behind the bar so you can decide whether to order another pint?

A few such projects can be found online. One of the most straightforward and earliest is Ismail Uddin's Science Exposure blog, **hsmag.cc/IsmailUddin**, where he explains how to make use of the live bus information for London buses (helped by them all having GPS trackers). The same principles can be used for National Rail services, as well as for London Underground. The project (and other similar ones such as **hsmag.cc/GitHubBus**) queries the countdown API and then pulls in data to use with Raspberry Pi. For a train-based version, follow this link: **hsmag.cc/TrainStation**.



You then need an LCD module to display information parsed from the live data feed. A dot matrix display completes the look. Just remember to throw the screen a glance so you can effortlessly time when you head to the station or bus stop.

CONSULT USER-GENERATED WEATHER MAPS

The British obsession with the weather is legendary, so it's perhaps no surprise that there are so many UK-originated DIY weather stations, including a Met Office project designed to engage children: hsmag.cc/metoffice.

The Met Office even maintains a Pinterest board for stylish ways to present your weather data: **pinterest.co.uk/metoffice.** Citizen science fans could also consult the weather map hosted by the Raspberry Pi Foundation, which sent out kits to schools (**hsmag.cc/weatherstation**) resulting in 1000 weather reports originating from around the UK. This is ideal if you simply want to pull in data from a crowd-sourced site.

If you're able to use this data as an API, you've got your very own bespoke weather forecasting system.

You could do a lot worse than follow the instructions at the aforementioned link to set up your own weather station. You'll need a Raspberry Pi; anemometer; weather vane; BME280 pressure, temperature, and humidity sensor; and a digital thermal probe; plus weatherproof enclosures to protect the electronics. A weather station forum provides additional ideas of how and where to mount your station, as well as how to pull in data from sites such as Weather Underground, a popular crowd-sourced weather site.

If you wish to, you can connect your weather station to the user-contributable Weather Observations Website at **wow.metoffice.gov.uk**, which is specifically designed for use by – and uses data from – homemade weather stations. →

Above
If setting up a weather station as a family project, consult the Met Office Pinterest board for ways to jazz up the output with infographics

Above left
Raspberry Pi provides an excellent tutorial on building your own weather station with an anemometer, weather vane, and associated sensors

Left
Weather and tide information becomes an object of beauty in the Weather Tide Project (hsmag.cc/WeatherTide)

NOT-SO-SMART LIGHTING

Smart light bulbs are a great idea if you want to be able to dim your lighting to set the mood, enjoy a bright welcome home, or give the illusion you're in residence and ward off possible intruders. Their ability to detect motion, as well as work to a timer or be triggered by low light levels, is certainly appealing.

However, as with home thermostats, the connectedness that gives smart light bulbs their smartness has caused some concern about privacy. As with many IoT devices, security company Check Point Software raised concerns about vulnerability in the Zigbee low-power wireless protocol that enables such products to connect to home networks via a bridge. This possible flaw was patched with a firmware update (if you have a Philips Hue bulb, go to the Settings menu to update it), but not before alarm bells had begun to ring about smart bulbs.

We don't advise trying to design your own light bulb, but ensuring you have a secure network for all of your IoT devices makes sense. Apple's HomeKit is a good choice for IoT devices that offers robust security, since it approves the devices that bear its 'works with Apple' badge and limits access to its hardware ecosystem (**hsmag.cc/HomeKit**). A smart bulb, such as a Yeelight, needs access only to your iPhone rather than your home network, so as long as you PIN/password-protect your smartphone, you should be in no danger.

Right ◆
Smart lighting linked to Apple's HomeKit network can be directly controlled by your iPhone

CREATE YOUR OWN SMART FRIDGE

If your idea of smartening your home involves a sentient fridge that intuits when you're running low on eggs, vegetables, or milk and adds those items to the shopping list, this low-tech, snooper-free alternative to an internet fridge may appeal: hsmag.cc/fridge. A webcam is trained on the fridge door and is set to take and share a photo whenever the fridge is opened. Doubling up as a cunning means of catching or deterring midnight snackers, the webcam is triggered by the fridge's light sensor which is connected to Raspberry Pi. To recreate this setup, you'll need a WiFi-enabled Raspberry Pi (or another model with a WiFi adapter), an ultra-wide-angle webcam, and a light sensor. You then need to set up Raspberry Pi and the webcam as a motion sensor to trigger the camera shutter, and to plug everything into a powered USB hub. Opening the door and letting in light is an effective trigger, while FS webcam software provides the single line of code to trigger the shot.

Since that original Hackster article was published, there's now a PIR motion sensor designed for exactly this, so you could just follow these instructions: **hsmag.cc/PIRSensor**.

Of course, the next stage would be to add some form of object recognition, along with a checklist of expected items in the fridge. Something like the KittenBot KOI (**hsmag.cc/KittenBot**) could be useful since it offers a straightforward means of photographing and tagging items for subsequent recognition. If you wished to add some form of automated ordering based on items that aren't present in your fridge, you'd need to have an IFTTT list ready, perhaps using Todoist (**todoist.com**), that triggers the ordering process.

The next step could take you back into the realm of the big tech companies – the Sainsbury's Alexa Skill offers Alexa-based grocery orders – or you could make use of Tesco's own IFTTT widget, **ifttt.com/tesco**, to automate your shopping list. Although this is set up to work with Apple or Android smartphones and apps, you can use an email login instead, then select which features to connect, including the option to add items from your Smarter shopping list to your shopping basket. Granted, all this isn't quite as neatly one-click as having an internet fridge do your grocery shopping, but you end up with much more control. →

Above
Add a wide-angle webcam to your fridge to monitor its contents, using Raspberry Pi to trigger a photo each time the door is opened

Below
The KittenBot KOI is a simplified AI device that can be quickly trained to recognise items and then register their presence, or otherwise, in your fridge

GROW YOUR OWN

Right Technovation designed and built a smart watering system that automatically waters their plants twice a day unless more or less is needed

Below A Grow HAT kit provides soil sensors for three plants and emits an alert if the soil gets critically dry

IoT systems that extend to the garden seem like a natural extension of the temperature and humidity sensors used to check the weather or home heating. As with other DIY monitoring devices, there are kits that provide an all-in-one setup – Pimoroni's Grow HAT and the Raspberry Shake measure soil moisture and earth tremors, respectively – as well as online instructions explaining how to create a system from scratch. Michael Klements from The DIY Life's Arduino soil moisture sensor project provides a great walkthrough: **hsmag.cc/SoilMonitor**.

As with the Grow HAT, it uses a traffic light LED system to let you know whether you're under- or overwatering your houseplants, making use of a capacitive soil moisture sensor probe and an Arduino Pro Mini. Depending on your needs, the system can be powered by a lithium-ion battery or, like Pimoroni's version, be mains-powered via a USB charging board and USB power bank. Michael provides useful details about adjusting the intervals between checking moisture content to extend battery life, as well as calibrating the sensor for accuracy.

Smart watering system

For larger-scale gardening projects, including hydroponic systems and controlled environments in which mushrooms and microbes might thrive, you could set up an automated watering system that can be controlled by smartphone.

This Instructable (**hsmag.cc/IOTGarden**) details how to set up a smart garden that keeps tight control on how much water is used, as well as monitoring plant health. Based around Google's Firebase and Raspberry Pi 3B with a GrovePi+ sensor kit, each plot in the IoT smart garden is an individual box that functions like a raised bed, with a sensor probe and a length of drip irrigation hose running up and down each row of plants. Makers Technovation (Kousheek Chakraborty and Satya Schiavina) bought items such as the solenoid valve to control the water flow from hardware and electronics stores. But they 3D-printed nozzles, pipe joints, and tube plugs, and made their own tarpaulin-lined planter boxes, keeping costs to a modest $50. They built a sensor box containing all the electronics from leftover plywood, and fitted it with hooks for each sensor probe. Details of each planter were added to a Google Firebase cloud-based database. MIT App Inventor was used to create the smartphone app. This displays humidity, moisture and temperature, and light level information for each growing environment and can be used to toggle the water pump on and off depending on readings. Left on 'auto', plants are watered twice daily at set times.

INTERNET OF GREEN THINGS

Above
Our slug problem means we have to grow our veg quite large in the greenhouse so they're strong enough to survive attacks by these slimy menaces when they're finally planted out

The growing season is now well upon us, and with it, the labours of keeping plants alive. Species that have survived unaided for millennia seem to die merely upon sight of this author's garden. And so, we turn to technology to help us. In this case, we're going to create a smart greenhouse – one that keeps an eye on the temperature and humidity, so we can make sure it doesn't get too hot or too cold.

To make this easy, we've opted to use a board that has everything we need on it: the FunHouse from Adafruit. This has an ESP32-S2 processor which has WiFi (letting us view our data without having to head over to the greenhouse), and an AHT20 temperature and humidity sensor. There's also a screen, so we can see the temperature if we are in the greenhouse.

We're going to use Adafruit IO as the data back end, as it's easy to use and free (for basic usage). You can sign up at **adafruit.io**. Data is structured into feeds which can then be displayed on dashboards.

We won't go through the code line by line because it's boringly straightforward, but you can grab it from **hsmag.cc/Greenhouse**. It reads the current humidity and temperature every ten seconds, and every 60 seconds it sends this data over WiFi to Adafruit IO. Additionally, if you press the middle button on the display, it will turn on the screen and you can see the latest information.

In addition, you'll need a secrets file that's structured like this (but with your information in it):

```
secrets = {
    'ssid' : 'WIFISSID',
    'password' : 'WIFIPASSWORD',
    'aio_username' : 'USERNAME
    'aio_key' : 'KEY',
    'timezone' : "America/New_York",
   }
```

All the upper-case words should be replaced with your own information.

The easiest way to set up the Adafruit IO side is to create a new dashboard. Click into the Dashboards tab, then select New Dashboard. Give it a name and you should be presented with a bare screen. The cog icon opens a menu with an option to add new blocks that display or accept data. Use it to add a line chart to the dashboard. You'll be asked to select the feed for this content block, and here you can enter the name of the feed. We've used gardentemp and gardenhumid, but you can use whatever you like (although they have to match the code). Once these are added, you should be able to start your code.

Notifications

We don't want to have to check the temperature constantly. Instead, we want a way of notifying us when things get too hot (so we can open the greenhouse roof to cool it down). The IFTTT web service lets you link two things together so that If something happens, it automatically triggers something else. In our case, if the temperature in the greenhouse gets too much, it sends us a notification. You'll need to sign up for a free account at **ifttt.com**. Then, in My Applets, click Create. Click Add to create the trigger mechanism. Select Adafruit, then monitor a feed. Set it so that if the gardentemp feed is greater than 35, it triggers.

Now you can select what happens if this applet triggers. Select Notifications, and then enter the details you want the notification to have. □

FEATURE

How I Made
MIDI FIGHTER

A Pico-based musical instrument with arcade buttons

By **Liz Clark**

MIDI Fighter-style controllers (MIDI controllers with grids of arcade buttons) have been a staple of the DIY MIDI controller community for years. This project continues that tradition with the Raspberry Pi Pico. A grid of 16 arcade buttons lets you play MIDI notes faster than you can yell "Hadouken!", either live with hardware or with your digital audio workstation (DAW) of choice.

The Pico is the perfect board for a project like this. With all the GPIO pins, you can directly wire your inputs and outputs without issue. The copious GPIO also allows this MIDI controller to have some special features. There is a screen with a GUI representing the 4×4 button grid, along with the assigned MIDI note numbers. Below the screen is a five-way navigation switch that allows you to select the individual buttons and adjust their MIDI note number on the fly rather than having to adjust the code.

Finally, there is an AW9523 expander board for control over the arcade buttons' LEDs. Both the screen and AW9523 are Adafruit STEMMA boards, which means they can be chained together to work over I2C.

Of course, there are some parts of this project that are optional, like the buttons' LEDs and the screen. Feel free to use this build as a jumping-off point for a simpler MIDI controller or change up some of the features to better suit your needs.

SUPPLIES

- 16 arcade buttons
- Raspberry Pi Pico
- Adafruit AW9523 STEMMA • GPIO Expander
- Adafruit Grayscale 1.5" • 128×128 OLED Graphic Display • Five-way navigation switch • USB micro-B extension cable • Silicon wire
- Soldering iron

INSPIRATION

Noé Ruiz and I love to collaborate on projects together. We've worked on a few MIDI projects in the past, and had discussed

Right
You don't have to paint your music studio bright pink

wanting to do a MIDI Fighter at some point. The MIDI Fighter-style controller also has a special place in my maker heart, since it was my very first DIY MIDI project that I worked on a few years ago. It had code written with Arduino, and was one of my first big soldering projects as well.

When the Raspberry Pi Pico came out, Noé and I thought it would be perfect for a MIDI Fighter, since it had so many GPIO. Noé also wanted to add another feature: the ability to change MIDI notes on the fly instead of having to edit the code. Noé is an accomplished beat drummer and felt that this feature would streamline his live drumming process when working on a beat with his DAW and software instruments. I was excited for the challenge to implement that feature and code up a user-friendly GUI. I think that it really makes this version of the classic MIDI Fighter-style controller stand out from the crowd.

3D-PRINTED CASE

Noé Ruiz from Adafruit designed the 3D-printed enclosure. It has three parts that snap together. The top has cut-outs for the arcade buttons and mounting points for the screen, five-way switch, and Raspberry Pi Pico. The Pico is also featured in the case, with space for a cut-out to have it displayed behind a piece of acrylic or a 3D-printed cover in clear filament.

There is a bracket that allows for the Raspberry Pi Pico and AW9523 to be mounted together so that the AW9523 can sit directly below the Pico. This comes in handy for wiring since the buttons' wires will all be running to the same area in the case.

The final touch for the case is a handle that sits at the top. It isn't just for aesthetics, though. When snapped in, it lets the MIDI Fighter sit at an angle that makes it more ergonomic for playing on a tabletop. Of course, this also makes it extra portable and gives it the feel of a retro lunchbox combined with a boom box.

In addition to being a beautifully designed case, it aids in the assembly for the project.

The top has cut-outs for the arcade buttons and mounting points for the screen, five-way switch, and Raspberry Pi Pico

FEATURE

Above
Need more buttons? Add a GPIO expander

Below
All the grounds get connected together

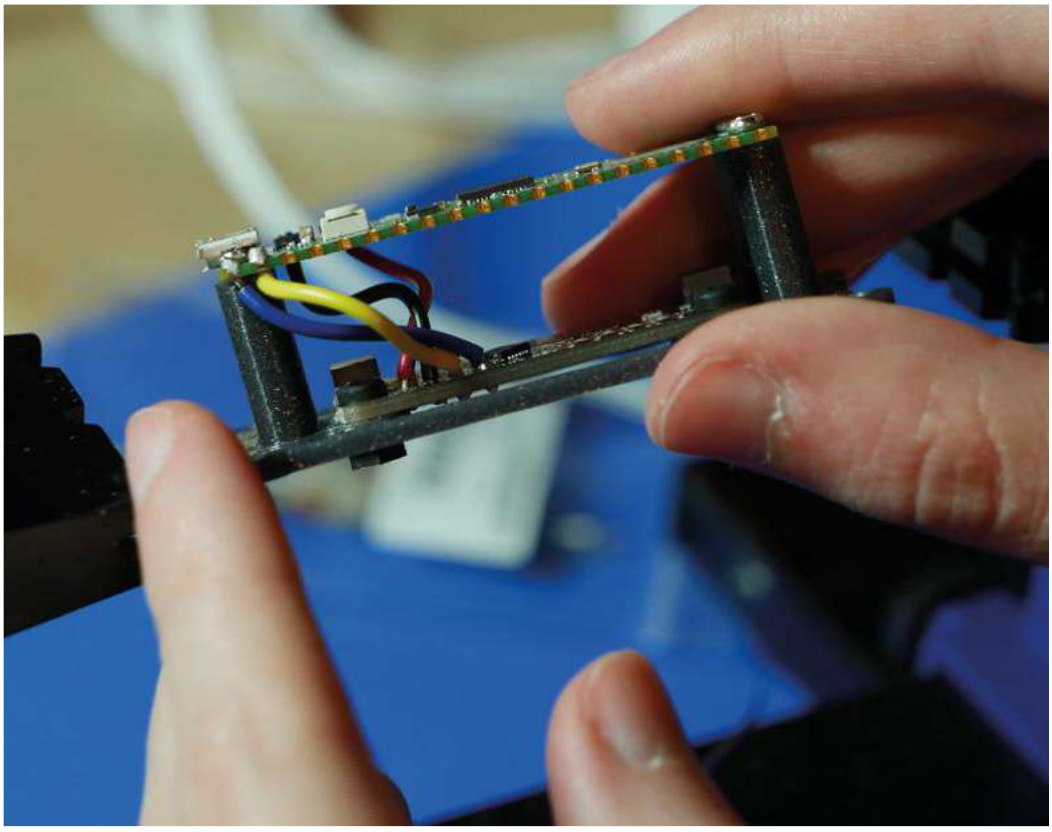
ASSEMBLY

1. After printing the case, pop the arcade buttons into their holes on the top lid. This allows them to sit securely while you solder everything up. Begin soldering by running a ground line between the buttons' inputs and LED grounds. This will keep your wiring nice and organised without a lot of bulk.

2. Next, attach the Raspberry Pi Pico and AW9523 onto their 3D-printed bracket. Cut four pieces of wire for the two I2C connections, power, and ground. Run these wires vertically between the two boards to continue to keep the wiring clean and compact, and then solder in place.

3. Running 16 wires for the button inputs and 16 wires for the buttons' LEDs for a total of 32 wires can get out of hand quickly. Before heating up the soldering iron, run the wires from the buttons to their pins to determine the correct length needed, while keeping enough slack so that the solder joints won't end up having too much stress. You can keep these wires organised with twist ties, or some tape in a notebook, so that you don't lose track of them. It may also be helpful to use different colours of wire for the button inputs and the LEDs.

4. After you have the wires cut to length, solder them from the buttons to the



Left
3D printed standoffs make assembly easy

The easiest way to solder to the I/O is to insert and hold the wire into the board's pins and secure it with a little solder

Pico and AW9523, respectively. Try to avoid getting any of the wires twisted. The easiest way to solder to the I/O is to insert and hold the wire into the board's pins and secure it with a little solder on the tip of your iron. You can add extra fresh solder after it's secured.

5. The five-way switch is a through-hole part. There's a PCB that acts as a breakout board for it and its design files are available so that you can order your own from a fab house or mill it. If neither of these options appeal to you, though, you can 3D-print a carrier board for the switch. It has the same mounting holes as the PCB and recessed holes for its legs so that you can solder to them without immediately melting the piece.

Like the arcade buttons, it's best to measure out the wires for the five-way switch ahead of time, and then solder to the switch's leads and the Pico's pins.

6. The screen connects to the AW9523 via a STEMMA connector, which means no soldering is required. Since the AW9523 and Raspberry Pi Pico are connected via their I2C, power, and ground pins, the screen is also connected.

7. After soldering, you can attach the screen, five-way switch, and Raspberry Pi Pico to the case with M3 screws. Then, take the USB micro-B extension cable and plug it into the Pico. Then, attach the socket to the side of the case with the included screws.

CODE

The code is written using CircuitPython, which has full support for Raspberry Pi Pico. The documentation and intuitive nature of Python make it achievable to code a hardware project like this, that includes a GUI with two splash screens, MIDI input, GUI navigation, controlling the arcade buttons' LEDs, and live assignment of MIDI note numbers, without too much stress. The full code is available for download from the Adafruit Learning System repository on GitHub: **hsmag.cc/CodeMIDI**.

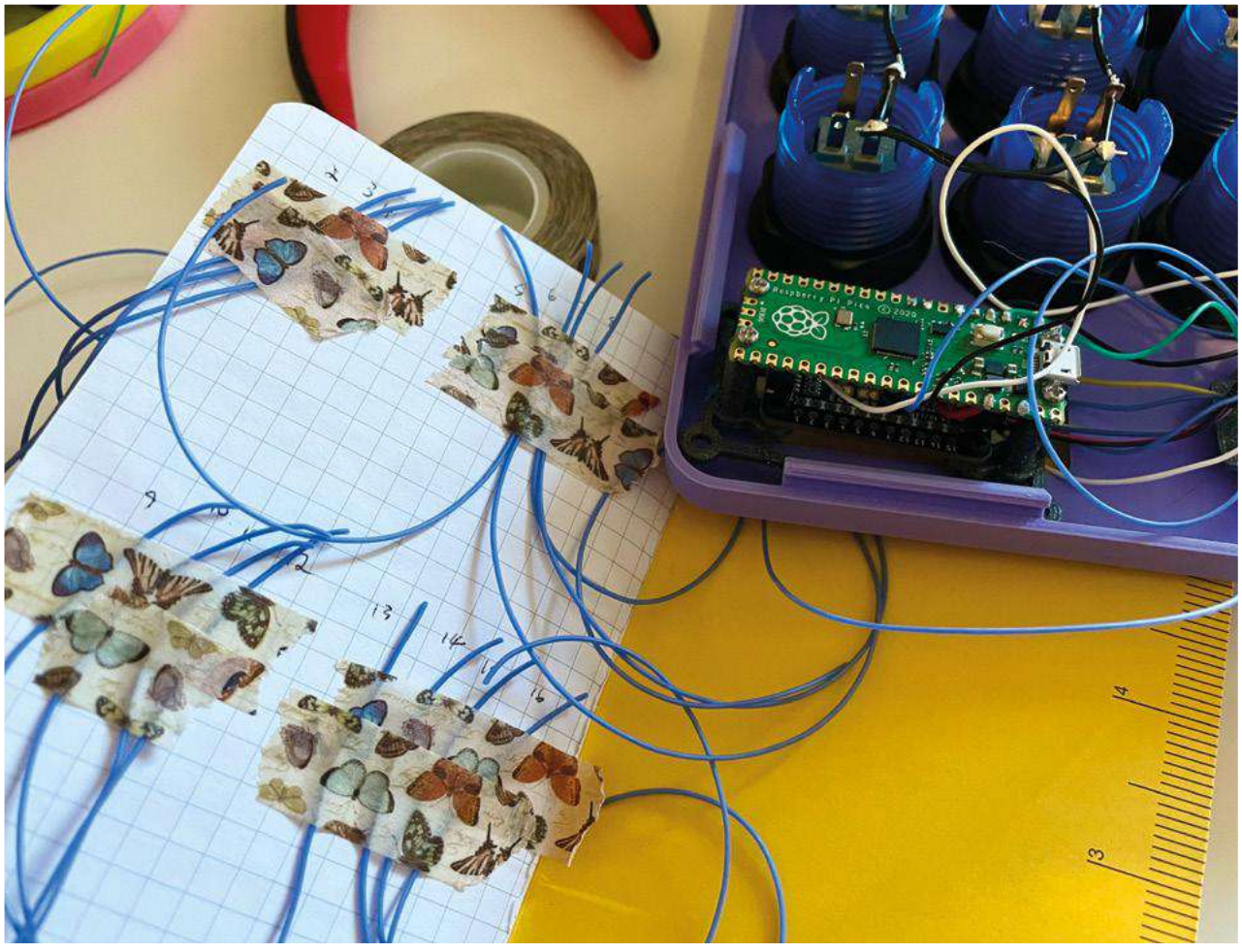
CHANGING THE MIDI NOTES

The screen's main GUI has a grid of circles meant to symbolise the grid of arcade buttons. Each circle also has a number in the centre, which signifies the current MIDI note number assigned to the corresponding arcade button. There is a square on the screen as well that outlines the currently selected circle. When you navigate the five-way switch, the square moves as well, acting as a cursor for the GUI. You can move up, down, left, or right, all around the screen. The navigation also wraps around the screen, meaning that if you, for example, press up when you are currently on the top row, the cursor will move to the bottom row.

The 'five' in five-way switch denotes a select option, meaning that you have a built-in push-button. When you press select, your highlighted circle will fill the screen with the current MIDI note in a much larger font to also fill the screen. This is considered the second GUI. The selected arcade button's LED will also blink to ensure that you have selected your intended button to adjust.

When you press up or right with the five-way switch, the number will increase, and if you press down or left, the number will decrease. You can press the arcade button to hear the changing MIDI note to test your new note selection. When you are satisfied with your choice, you can press select again with the five-way switch and return to the main GUI

The Raspberry Pi Pico MIDI Fighter has features that make it ideal for both playing live and noodling at home working on a track



grid of circles. Your new MIDI number will now be displayed in the smaller circle in the grid.

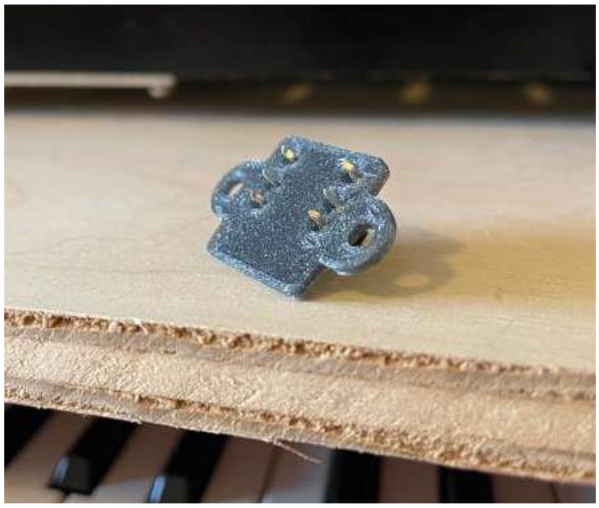
USAGE

The Raspberry Pi Pico MIDI Fighter has features that make it ideal for both playing live and noodling at home working on a track. You can use it with your computer's DAW over USB and change the assigned MIDI notes on the fly for different settings or to different note input options. The fast and accurate nature of the arcade buttons makes them great candidates for live beat making and performance.

If you prefer to make music without a computer, you can use a MIDI host device to connect the Pico MIDI Fighter to other MIDI devices. There are speciality hardware devices with these features, or you can build your own, such as with a Raspberry Pi 3B, 3B+, or 4B since they have built-in USB ports. Personally, I use mine with my Raspberry Pi 3B MIDI host connected to a Winterbloom Sol Eurorack module, which converts MIDI message to CV signals for modular synths.

Whichever your preference, this is a fun build for any music enthusiast that makes use of all of the Pico's many GPIO. If music isn't your thing, it would be a fairly painless process to convert the CircuitPython code to be an HID macro keyboard instead.

If you're interested in a more in-depth, step-by-step build, a tutorial is available on the Adafruit Learning System by myself and Noé Ruiz: **hsmag.cc/MIDITutorial**.



Right
With a 3D printer, it's easy to make little mounts

Below
Let's make some rhythmical beeps and boops

HackSpace magazine meets...

Joel Telling

Going viral, in a good way

Joel Telling has been labelled as 'The Nicest Guy on YouTube/in 3D Printing/ in the maker community/in the WORLD' by an awful lot of people, and it's not hard to see why. He's enthusiastic, kind, and the sort of person you instantly warm to as he excitedly tells you about other makers and initiatives he thinks you should follow. For every 100 words of conversation about himself, he has another 200 words of praise about someone else.

On his YouTube channel, 3D Printing Nerd, Joel reviews an array of 3D printers and tools, prints impossible-looking models, and offers tutorials to better your understanding of the technology. And why? Because he's exactly what he says he is – a nerd. And we can't get enough of it.

Alex Bate tracked Joel down to find out what it took to become a 3D printing YouTube sensation. →

Above
Abusing 3D printers for fun and profit

HS You released a video four years ago, telling your followers you had quit your job at Adobe to focus full-time on your channel. It was the first of your videos I'd ever watched, and I remember thinking, "Wow, that's huge!"

JT It was huge. Part of what made it so big is that I have a wife and three kids. And here in the States, our healthcare's different from other parts of the world. And so part of quitting meant I had to have enough money so my kids could eat, and my wife could eat, and we could maintain the level of living that we currently had, and still afford to go to a doctor.

And what was great is that I had been monetised way before I quit Adobe. So I had money coming in. I just opened a business account at a bank and the money went into there. And I used it to help do business things, but also every once in a while, I'd be like, "Come on kids. YouTube's paying for this ice cream tonight." It was just fun. I mean, looking back, it's weird because now YouTube pays for everything.

That was what was on my mind. And the night prior to me handing in my notice at Adobe, I talked to my wife and she was like, "Are you gonna do it?" And I'm like, "I think I could do this. I think we could do this." She's like, "I'm behind you."

And part of what made it possible was a runway, and I know not everybody has this, I had a runway. And so the idea was I had six months to see if I could make this work. I wanted to give it an honest to goodness six months to try it out. Because I think that's enough time to figure out whether or not your plan is going to work. And I had that, I had that much runway – if I brought $0 in, I knew that we could maintain our current way of living for six months.

It went a little bit longer than that, obviously, but also, it wasn't great because I was getting these crazy headaches, and I was working my Adobe job and YouTube was becoming more.

Content creation itself is quite a laborious task. So, I was working a lot and not sleeping enough. And [I told] my neurologist, I was having headaches – I started to forget words, you know, that sort of thing.

It's crazy, like a medical professional, my doctor, my brain doctor, he said, "You should quit your job." And I was like, "Really? You think?" and he said, "Yeah, that's gonna alleviate a lot of things."

So I took medical advice and proper planning, and quit my job. And then I put out that video just to let everybody know. And it was a thing. It was definitely a thing.

HS There are younger creators nowadays who have never had a 9-to-5 job, who have been able to support themselves through social media from the start. But then you've got people like you who have a house, and three kids, and a wife, and pets and everything else. And you made that choice, even now when it's possibly not as easy to make decent money through AdSense on YouTube.

> There's a content creation event that happens and if you can capitalise on it, then it does become your career

JT It's not as easy. I don't know if AdSense is paying less, but through the progression of time, we have so many people creating so much awesome content that you either have to find a niche market, or you have to rise above the crop. You have to be the cream of the crop and get seen by more. So just as an example, my buddy, his name is Bender, he's on the radio here in Seattle and his son out of high school, went to college, didn't have a job, but he loved making videos, and he loved After Effects and doing stuff. And one day he made a video where he got a bag of mini babies from Amazon. And he put them all over his college campus in weird, crazy, random places. And he filmed it. For some reason that video went viral and got millions and millions of views. And that kicked off his channel. And so, like you're saying, there are people who've never had a job that can do this. It's almost like an event. There's a content creation event that happens and if you can capitalise on it, then it does become your career. Well, I mean, just like iJustine, right? Way, way back in the day, she got that hundreds and hundreds of pages AT&T phone bill for the iPhone, and she's going through it and talking about it. And I believe that that massive event helped launch her into the tech person she is today.

HS Do you think you've had your massive event?

JT A couple, yeah, I've had a couple. In content creation, obviously you look for a viral video, right? Something that takes off. Destiny, the video game, was doing really well and I put together one of the weapons from the game, just kind of assembling it in my garage. I had a couple of different angles. I did some fun tricks with the camera, and for some reason, my videos were getting five to ten thousand views, and that one got 180,000, which was pretty nuts at the time. It was my first, and then I'm sure you're aware of the hairy lion...

HS I have it right here in my notes. 'The massive blue hairy lion!'

JT His name is Bluefusa.

HS Bluefusa? Brilliant.

JT Yeah, there's a little titbit you get to know: his name is Bluefusa. I just thought it was cool. There were a bunch of people printing it at the time. It was the first

Below
Early on, Joel was known for his large 3D prints

time people were doing a sacrificial wall with strings of plastic in between so they could create the hairs or the fuzzy stuff. I had the creator of the original one make it larger, so that it worked for the large format. If I just scaled up the small one, the hairs would have been really thick. And so I wanted the model scaled up, but the hairs to be the same. I knew was gonna take a while, but I had a big machine and I was known for long prints – and so I set it off and 99 hours later, I'm there with a bread knife just kinda cutting away that sacrificial ball, and Bluefusa was born. I have no idea why that resonated with people. But that shot up like a rocket and it's over 3 million views now. I don't know why. It's just crazy.

HS You got your first 3D printer for Christmas. A FlashForge, I think.

JT The FlashForge Creator Pro in 2014.

HS Why did you want one? What made you ask for it?

JT Well, I'm a giant nerd. I mean, I've always liked nerdy things like ever since →

Below
Daft Punk, you will be missed

I was a kid. I take things apart; I put them back together. I used a lot of tape. I always wanted one and I had friends who had had some. I expressed that to my wife and kids, of course, but it wasn't in the budget; it was quite expensive.

I'd always watched videos from Barnacules, and Thomas Sanladerer. Those were the two. I'd seen stories on Slashdot, and those types of websites, but just seeing Tom and Barnacules, watching them get excited about their prints, and Tom being the engineer type to dive technically into it like that, they really started my brain firing off. So, when the FlashForge was on sale online, my wife and kids bought it and I opened it up on Christmas. I was up until two o'clock in the morning just futzing with it because I had a new thing that could make stuff. And my first thought was, "I'm going to make money." That was my first thought. And so I went about 3D-printing a bunch of cookie cutters.

I would design my own, or find some where the licence allowed for it. And I would sell them on Etsy for like five bucks apiece, or locally to friends. For example, here in Seattle, we have the Seahawks, an American football team. And I made a cookie cutter that was two pieces; you could cut out the shape and then use a second piece to imprint the design on the cookie so as it cooked, it looked like the logo. And yes, I was infringing on copyright logos, I'm sure. But the Seahawks have been typically pretty relaxed as far as people creating handmade content and selling it. You know, if you're knitting something and you're selling a dozen of these things, then they're really not going to come down on you. But if you make a product that is eerily similar to something they sell themselves and you're cutting into their business, then the Ban Hammer of the NFL will come down and hit you really hard.

So I was small potatoes. They weren't worried about me. So I sold a couple hundred cookie cutters to friends and family and on Etsy.

It was fun, but I didn't want a garage full of 3D printers just making things to sell, you know. I was very interested in the technology. I have a GoPro and I usually use it at Easter as the kids search for eggs. I'm running around and the footage is all wobbly, so I thought what I needed was a two-handed rig where I could attach the GoPro. OpenSCAD is the closest thing to JavaScript, and I knew JavaScript, so I started using it to design a rig. And it was awesome. I enjoyed the problem-solving and the process of creating and testing to see if the solution worked. I thought others might enjoy it too. So I started to turn on the camera and record stuff. And I'd always had cameras because I liked pictures. And I liked video and I used to film my friends doing stuff just for the fun of it. Right? So I thought I could film myself. I turned the camera on and I started talking about the process and what I was doing, and then people got interested. And they started subscribing and leaving comments. I started replying and answering questions and making videos about what I was printing. And that's literally how it kicked off. My yearn to teach is what made a channel grow. I just wanted to show people cool stuff. Isn't that what content creation is all about? Just showing people cool stuff you're passionate about. I just happened to be, at the time, passionate about wanting to create a solution for a problem I had using the tools at hand.

> My yearn to teach is what made a channel grow. I just wanted to show people cool stuff

HS Are there other members of the 3D printing community that helped your growth as a 3D printer user?

JT At the time, those are the two that I remember. I know Angus over at Maker's Muse started around a similar time, or at least started growing around a similar time. And I got to know Angus, and his content has always been great. I always credit his Fusion360 tutorials on getting me kick-started on using the software. I have put up my own Fusion360 tutorials, like for fidget spinners, and a rainbow model I made just recently. So, along the way, I've watched a lot of content creators, whether it's 3D printing or not. Through my journey, there have been a lot of interactions I've had with a lot of different people who have made cool content.

HS The maker movement online has definitely grown over the last few years, with more and more people owning their own 3D printers. What do you think has attributed to this growth? Cost? Accessibility? Creators like yourself making them more popular?

JT I think we're at a point where we have a price that has democratised printing for a greater number of people than ever before, the Creality Ender Series, ELEGOO has the Neptune, etc. We're are a point where the price is low and the quality is, I don't want to say high, but the quality has finally hit above a certain bar that has made it worthwhile for the non-super-nerds. And now it's up to people like myself and others to publicise that and talk about how well it works for us. What are the problems with it? And how do you fix these things?

You have Chuck Hellebuyck's Filament Friday putting out videos about either what he's done to use Cura to do something cool, or how he's fixed or upgraded one of his Ender 3D printers. Chuck's a massively nice guy, he's like a brother to me.

HS This may be an impossible question, but do you have a favourite 3D printer?

JT It's the one loaded with filament next to me, ready to go and printing my favourite model. →

HS Ha ha, that's fair.

JT While I have lots and I like a lot of my machines, there are specific tasks that some of them are better at than others. So, if I want to print some ABS right now, I'm going to go use my Raise3D E2 because it has an enclosed build chamber and the extruder works well for ABS, or, if I just want to throw some PLA down and just get a couple quick prints out, well, I'm going to load up one of my Prusa machines, or my CR-6, or the new Ender 3 V2 sat right over there, right? The FlashForge Creator Pro could be considered my 'favourite' because it was my first, and that's what made all this possible. But then, the Prusa i3 MK2 was my first Prusa printer, which led to me meeting Joe and getting to go to Prague and touring the factory. Or, the CR-10 could be a favourite printer as it helped the channel and helped Creality grow, and made it possible to get other machines to tell people about. It's like, "who's your favourite child?" Well, I like them all.

HS Are there any printers you haven't played with yet that you want to? Is there anything on your radar that you'd like to get your hands on?

JT What I want to do is start showcasing more of the professional and the industrial side of 3D printing. The interesting thing is, if you think about it, kids in class, or kids going to university, if they want to work in 3D printing or work in manufacturing, it's not like they're going to go to Ultimaker. Most likely, they're going to work for a large company which has multimillion-dollar machines making parts. So, why not showcase that side of the industry in the same way we showcase these cool consumer things? Why don't we make industry cool and appealing to a younger crowd? Because that younger crowd, if they go through their schooling in a way that makes them able to work in that industry, that's where they're going to work.

If you look right now, a lot of the industrial side of 3D printing are creating flashy corporate videos that look like flashy corporate videos. And they're decent, right? Technically, they're good. But there's a lot of times you're not going to appeal to a younger crowd. And I think that's a misstep. And so, what we want to do is supplement their offering in a way that appeals to a younger crowd and gets that younger crowd excited for the professional side of things.

[At the time of the interview, Joel was 24 hours away from a YouTube live stream with the actor Neil Patrick Harris. Watch it here: **hsmag.cc/JTelling**]

HS So, a very exciting thing is happening with a certain Neil Patrick Harris.

JT That should be fun. A year or more ago, Neil sent out a tweet. And I think he was looking for advice on 3D printers. He has 26 million followers on Twitter, so he got a number of replies from companies, individuals, everybody tweeting at him. But a large portion of people actually said "talk to Joel". I told my producer David about this. And David has worked in LA for 20 plus years. He's done a number of TV shows, and streaming shows. He has a friend who knew someone on Neil's team who put him in touch with Neil's assistant. And a year later, David says, "We have a Zoom call set with Neil and his assistant. You ready?" Ummm! So, we log in and sure enough, there's his assistant, there's Neil, there's me, and there's my producer. And that was my first official chat with Neil about this. He's just a cool guy.

We wanted to stream something. He's got a Prusa and we're going to do an unboxing and we have two or three projects planned in the future, which is cool. Neil's going to tweet about it and I've never had anything I've said reach 26 million people!

We're always looking for ways to, I don't want to use the word legitimise, but legitimise what we do to a mass audience, right? How do you take a niche thing that Adrian Bowyer (a very prominent member in the history of consumer 3D printing) did and encapsulate that into a way that's consumable for a mass audience? And I think we're getting there. So, I get Neil Patrick Harris for two hours in a live stream on my YouTube channel. I'm still not comprehending that fully, but, I think there's power in that, not just to get a couple more subscribers but there's power, because there's people that know of Neil that have never seen me, and they're going to see him working with the machine, producing a thing and showing it off, and think 'wow', and go in search of others like me and be excited. They'll talk to their parents, their grandparents, their loved ones. I think

that's how we get a more mass adoption of this technology – is we get famous, well-respected people who are willing to give it a try. And I think we're doing OK right now.

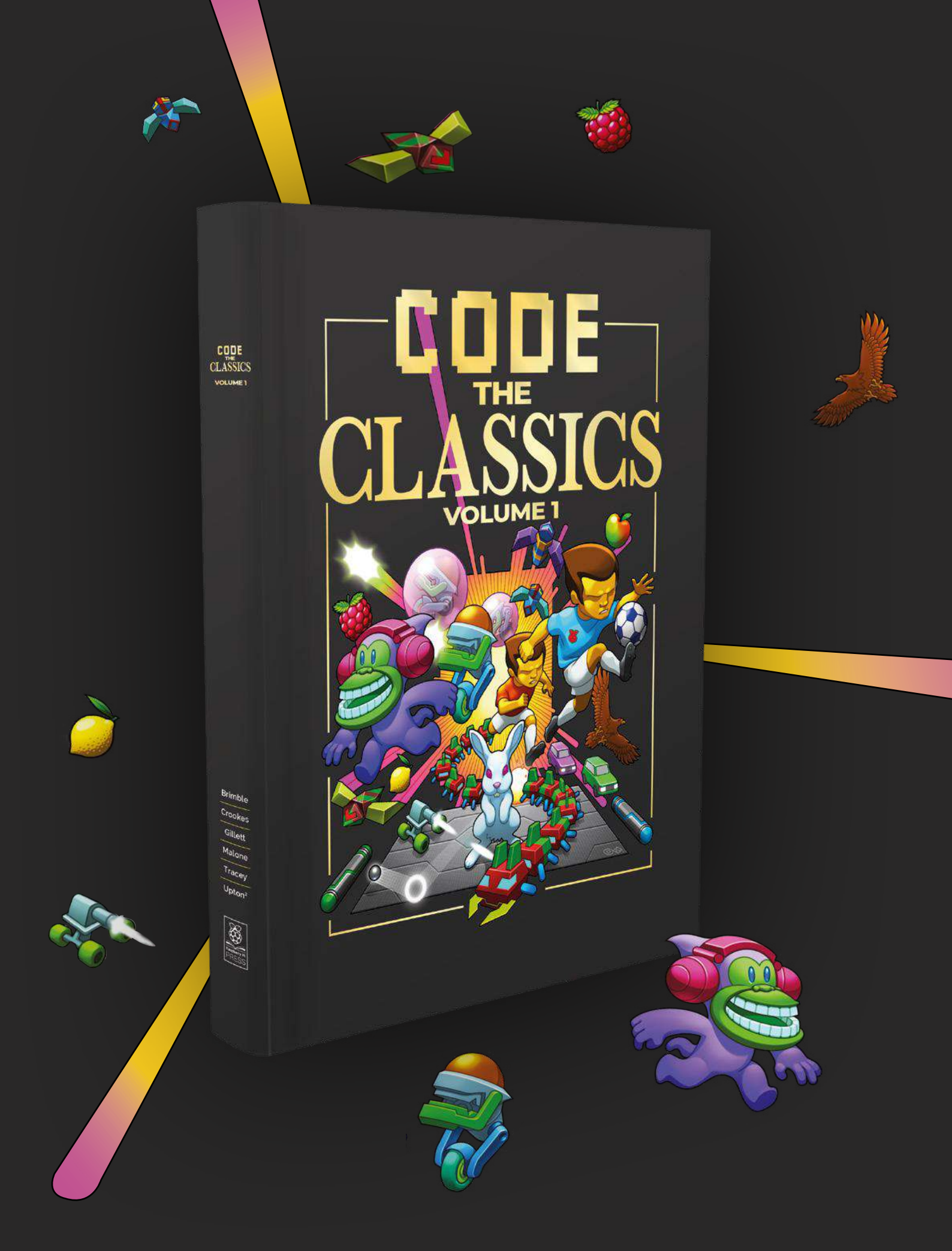
HS I'm gonna imagine you get an awful lot of people who tell you that they've bought a 3D printer because of you.

JT I do. Yeah, I do. Or they'll say they bought their 3D printer because they watched me and Maker's Muse or Chuck Hellebuyck, and they saw me on his channel. So they watched me and now they have a printer. It's neat how these rivers kind of flow and get the people excited about this. There's a lot more people with 3D printers now than there were just twelve months ago.

> "I just enjoy being able to have a reach, because I hear from people that they are inspired by the stuff that I'm showing"

I just enjoy being able to have a reach, because I hear from people that they are inspired by the stuff that I'm showing. It's weird, if you think about it, right? Someone could watch one of my videos and because of that, it sets off a chain of events that makes them own a company about this stuff. Or perhaps I mentioned something and it resonates with someone and then they go into the medical field and we have a disease cured. That's what we need to do, right? To get these wonderful moments in the future, we have to provide all sorts of loving, caring inspiration for all of these people who just are interested and curious and want to make cool stuff. And so, I always stress to people not to be afraid to fail, and to tag me in their stuff. And so hopefully we plant the seeds and we grow the flowers and we bear lots of really incredible maker fruit.

Follow Joel Telling on YouTube here: hsmag.cc/3DPrintingNerd, and Twitter at twitter.com/joeltelling. □


CODE
THE
CLASSICS
VOLUME 1
Brimble
Crookes
Gillett
Malone
Tracey
Upton²

SUBSCRIBE TODAY

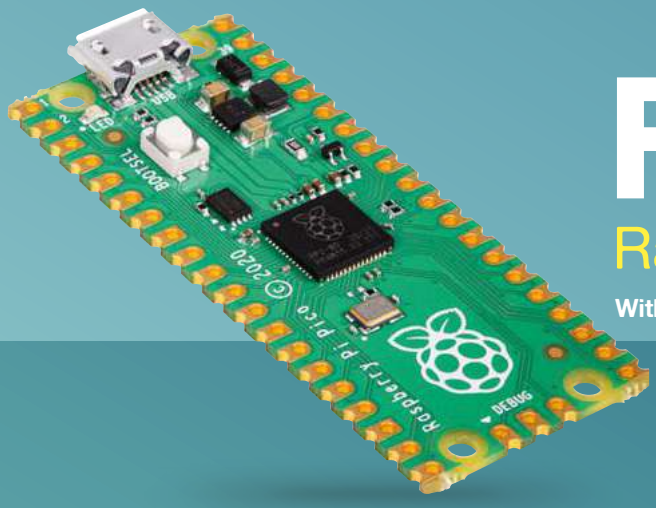
Get 12 issues of HackSpace magazine delivered to your door for just

£55 (UK)

£90 (USA)

£80 (EU)

£90 (Rest of World)



FREE!

Raspberry Pi Pico

With your first 12-month print subscription

This is a limited offer. Not included with renewals.
Offer subject to change or withdrawal at any time.

Subscribe online: hsmag.cc/subscribe

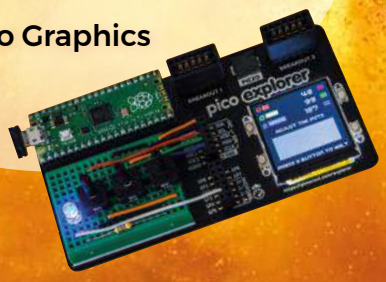
Subscription starts with next issue

HackSpace
TECHNOLOGY IN YOUR HANDS

FORGE

HACK | MAKE | BUILD | CREATE

Improve your skills, learn something new, or just have fun tinkering – we hope you enjoy these hand-picked projects



Colours and characters on Pico

Use MicroPython to display complex graphics on a Pico Explorer

Tony Goodhew

Tony is a retired teacher of computing – he has been helping people to code since 1968. He started with FORTRAN IV on an IBM 1130 with Hollerith cards for input.

The Pico Explorer from Pimoroni is a great device for testing out different bits of hardware to Pico. There's a screen and four buttons on board, and you can hook up extra hardware with the small breadboard, pin headers and two breakout slots. The breadboard looks small, but with care, you can connect up several components. We've added three 10 kΩ potentiometers and an LED protected with a 470 Ω current-limiting resistor.

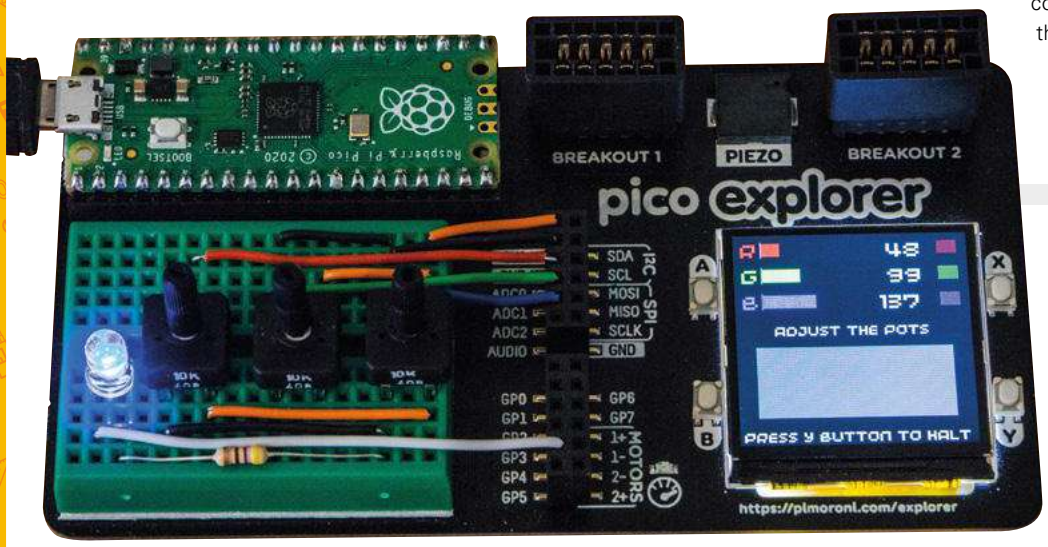
Here the orange wires are fed from the 3V3 socket, and the black wires connect to GND. The red, green, and blue wires connect the wiper pins of the potentiometers to ADC0, ADC1, and ADC2, respectively. The potentiometers fit across the centre gap, over the vertical wires, with the outer legs joined to 3V3 and GND at the bottom, with the single leg (wiper) at the top. The white wire connects GP4 to the longer leg of the LED (anode). The 470 Ω current-limiting resistor joins the cathode to GND.

Pimoroni provides an instruction to read the analogue-to-digital converter (ADC) pins.

```
red = display.get_adc(0)
```

This supplies a floating-point value between 0 and 1. Unfortunately, the ADC pins on a Pico seldom give zero at the bottom end or a reliable and stable 1 at the top end, so some of the code tidies up the readings to give full ranges: (0–100), (0–255), and (0–65535), as needed.

The program displays red, green, and blue values as a bar graph, with lengths in the range of 0 to 100 pixels. The corresponding colour values are displayed numerically and as small coloured rectangles. The large rectangle shows the colour resulting from combining these three colours. The average brightness of the three component colours is displayed as the intensity of the LED using PWM with the duty value in the range of 0 to 65535.



Below
The complete circuit, with an LED and three potentiometers

```
def showgraph(s,v,drop,rr,gg,bb):
```

- **s** is a single-character string label for the graph – R, G, or B
- **v** is a value in the range of 0 to 100 for the length of the bar – a percentage
- **drop** is a pixel count to push the displayed bar graph further down the screen
- the last three parameters (**rr**, **gg**, **bb**) define the colour of the bar in the range of 0 to 255

You can download the MicroPython program here: **hsmag.cc/issue43**

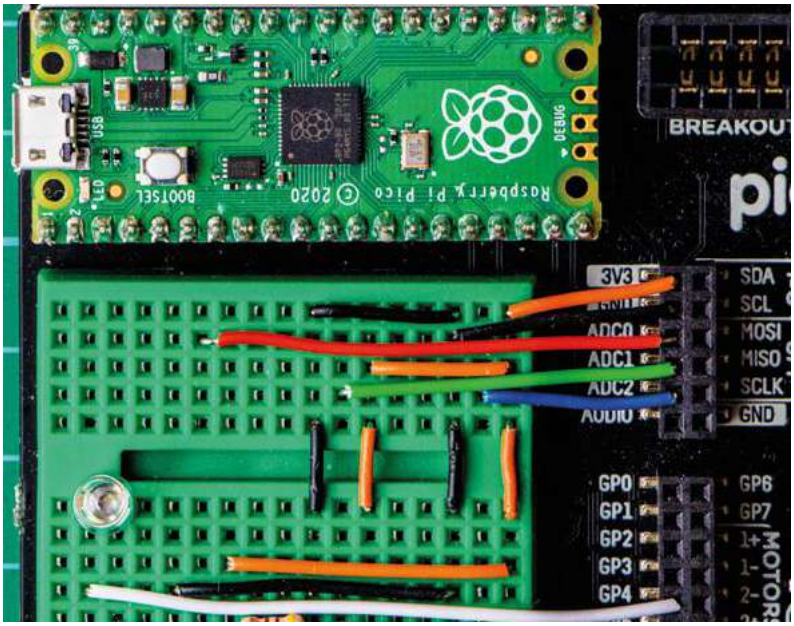
```
# Physical Computing with Graphics on Pico
Explorer
# 10K Ohm potentiometers on ADC pins
# LED with protection resistor approx 470 Ohms on
GP4
# Tony Goodhew 4th March 2021 (Tested on Vers:
0.0.8)
import picoexplorer as display
import utime
import machine
width = display.get_width()
height = display.get_height()
display_buffer = bytearray(width * height * 2)
display.init(display_buffer)
led = machine.PWM(machine.Pin(4))
led.freq(1000)

def blk():
    display.set_pen(0,0,0)
    display.clear()
    display.update()

def vert(l,t,b):   # left, top, bottom
    n = b-t+1      # Vertical line
    for i in range(n):
        display.pixel(l, t+i)

def align(n, max_chars):
    # Aligns string of n in max_chars
    msg1 = str(n)
    space = max_chars - len(msg1)
    msg2 = ""
    for m in range(space):
        msg2 = msg2 +" "
    msg2 = msg2 + msg1
    return msg2  # String - ready for display

def showgraph(s,v,drop,rf,gf,bf):   # Bar graph
    display.set_pen(rf,gf,bf)
    display.text(s, 8, 8+drop, 240, 3)
    display.set_pen(0,0,0)        # Blank old bar
graph
    display.rectangle(29, 10+drop, 240, 20)
    display.set_pen(rf,gf,bf)     # New  bar
graph
    if v == 1:vert(29, 10+drop, 25) # Ensure
these show up
    if v == 2:vert(30, 10+drop, 25)
    display.rectangle(29, 10+drop, v, 15)
```



Above
With the potentiometers removed, you can easily see the wiring

```
    display.set_pen(100,100,100)  # Base line
zero
    vert(28, 8+drop, 26+drop)
    display.set_pen(200,200,200)
    # convert v to colour value (0 .. 255)
    num = int(v*2.55)
    display.text(str(align(num,4)), 140, 9+drop,
240, 3)
    display.rectangle(210,9+drop,28,15)

# Mix colours
display.set_pen(200,200,200)
display.text("Adjust the pots", 40, 105, 230, 2)
display.text("Press Y button to halt", 5, 225,
230, 2)
running = True
while running:
    # Calculate percentages from pots
    rpot = int(display.get_adc(0) * 101)
    gpot = int(display.get_adc(1) * 101)
    bpot = int(display.get_adc(2) * 101)
    #Draw bar graphs using percentages
    showgraph("R",rpot,0,255,0,0)
    showgraph("G",gpot,30,0,255,10)
    showgraph("B",bpot,60,0,0,255)
    # Calculate colour values from pots
    rv = int(display.get_adc(0) * 255)
    gv = int(display.get_adc(1) * 255)
    bv = int(display.get_adc(2) * 255)
    # Clear old RGB colours
    display.set_pen(0,0,0)
    display.rectangle(210,9,40,90)
    # Draw new RGB colours
    display.set_pen(rv,0,0)
    display.rectangle(210,9,20,15)
```

```
    display.set_pen(0,gv,0)
    display.rectangle(210,39,20,15)
    display.set_pen(0,0,bv)
    display.rectangle(210,69,20,15)
    #Draw mixed colour
    display.set_pen(rv,gv,bv)
    display.rectangle(20,130,200,80)
    # LED shows average brightness
    duty = int(250 * (rv+gv+bv)/3 -300)
    # Adjust for pot errors
    if duty < 0:
        duty = 0
    if duty > 65536:
        duty = 65500
    led.duty_u16(duty)
    display.update()
    # Finished?
    if display.is_pressed(3): # Y button is
pressed ?
        running = False

# Tidy up
blk()
led.duty_u16(0)
display.set_pen(200,0,0)
display.text("All Done!", 55, 40, 200, 3)
display.update()
utime.sleep(2)
blk()
```

DEFINE CHARACTERS/ICONS

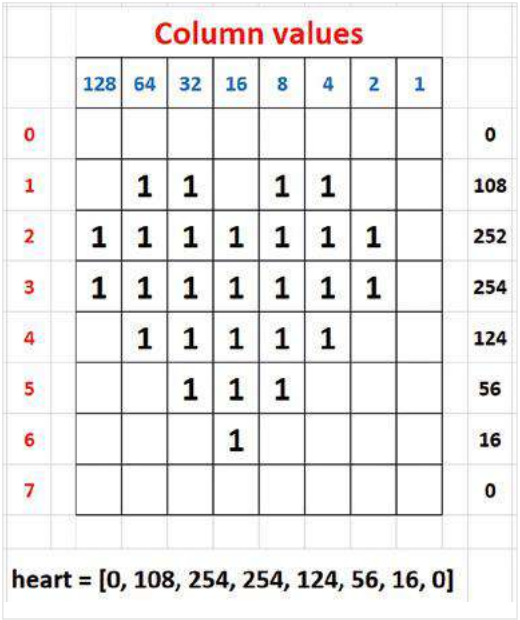
The second program demonstrates how to display extra characters or small icons on your screen.

An 8×8 grid is used to design the icon and code the shape. For each row, where there is a 1, you add up the values at the top of the column. This codes a simple heart shape. The eight numbers are stored in a list, with the top value first.

Since the pixels on the Explorer screen are so small, I provide two routines to place them on the screen at double and triple the natural size – **`mychar2()`** and **`mychar3()`.** The user provides values for the top left-hand corner of the positioned icon and the icon name. The script has definitions for a heart, a broken heart, a duck, and a bell. You could add definitions for arrows or the degree character for temperature displays. The program demonstrates a beating heart and swimming ducks.

You can download the program from our website: **hsmag.cc/issue43**

```
# Pimoroni Pico Explorer Icons
# Tony Goodhew 4th March 2021
```



Above
Characters can be defined as lists of 8-bit numbers

```
# Tested with Pimoroni UF2 Ver: 0.0.8

import picoexplorer as display
import utime
from machine import Pin
width = display.get_width()
height = display.get_height()
display_buffer = bytearray(width * height * 2)
display.init(display_buffer)

def blk():
    display.set_pen(0,0,0)
    display.clear()
    display.update()

#Defined icons/characters
heart = [0, 108, 254, 254, 124, 56, 16, 0]
b_heart = [0, 108, 222, 238, 116, 56, 16, 0]
duck  = [0, 96, 224,32,127, 254,124,0]
bell = [0, 24,60,126,126,126,255,12]
bits = [128,64,32,16,8,4,2,1]  # Powers of 2

def mychar2(xpos, ypos, pattern):  # Print defined
character
    for line in range(8):          # 8x8
characters
        for ii in range(8):
            i = ii
            dot = pattern[line] & bits[i]  #
Extract bit
            if dot:  # Only print if a 1
```

```
                display.pixel(xpos+i*2,
ypos+line*2)
                display.pixel(xpos+i*2,
ypos+line*2+1)
                display.pixel(xpos+i*2+1,
ypos+line*2)
                display.pixel(xpos+i*2+1,
ypos+line*2+1)

def mychar3(xpos, ypos, pattern):  # Print defined
character
    for line in range(8):          # 8x8
characters
      for ii in range(8):
        i = ii
        dot = pattern[line] & bits[i]  # Extract
bit
        if dot:  # Only print if a 1
          display.pixel(xpos+i*3, ypos+line*3)
          display.pixel(xpos+i*3, ypos+line*3+1)
          display.pixel(xpos+i*3, ypos+line*3+2)
          display.pixel(xpos+i*3+1, ypos+line*3)
          display.pixel(xpos+i*3+1,
ypos+line*3+1)
          display.pixel(xpos+i*3+1,
ypos+line*3+2)
          display.pixel(xpos+i*3+2, ypos+line*3)
          display.pixel(xpos+i*3+2,
ypos+line*3+1)
          display.pixel(xpos+i*3+2,
ypos+line*3+2)

blk()
display.set_pen(200,200,0)
display.text("Defined characters", 20, 10, 240, 3)
display.set_pen(200,0,0)
mychar2(20, 60, heart)
mychar3(50, 60, heart)
display.set_pen(0,0,200)
mychar2(20, 85, b_heart)
mychar3(50, 85, b_heart)
display.set_pen(200,200,200)
mychar2(100, 60, duck)
mychar3(130, 60, duck)
display.set_pen(250,180,0)
mychar2(100, 85, bell)
mychar3(130, 85, bell)
display.update()
# Beating heart
for n in range(6):
    display.set_pen(230,0,0)
    mychar3(50, 60, heart)
    display.update()
    utime.sleep(0.5)
    display.set_pen(90,0,0)
    mychar3(50, 60, heart)
    display.update()
    utime.sleep(0.5)
display.set_pen(200,0,0)
mychar3(50, 60, heart)
display.update()
# Swimming Ducks
display.set_pen(100,100,100)
display.text("Press button Y to halt", 5, 180,
230, 2)
display.set_pen(30,40,200) # Water
display.rectangle(0,150,240,18)
y = 130
running = True
while running:
    for xx in range(260, -50, -1):
        display.set_pen(0,0,0) # Wipe old ducks
        display.rectangle(0,y,240,21)
        display.set_pen(200,200,200) # New ducks
        mychar3(xx,y,duck)
        mychar2(xx + 35, y + 7,duck)
        display.update()
        if display.is_pressed(3): # Y button is
pressed ?
            running = False  # finish 'while' loop
            break # out of 'for' loop
blk()
display.update()
```

With slight modifications, to take account of the smaller screen size, these techniques can also be applied to the Pico Display board. □

Below
A range of new characters for you to play with

TUTORIAL

CAM and cuts – the CNC 3018 Pro

Putting our budget CNC machine through its paces

Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

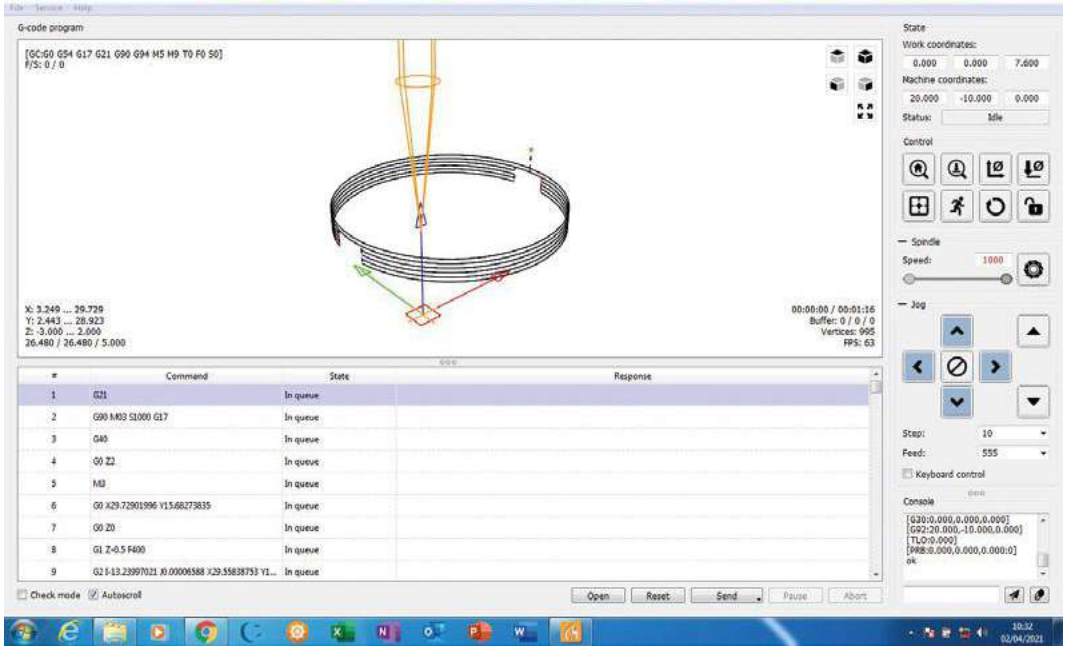


Figure 1 The GRBL Candle control software supplied with the CNC 3018 Pro, ready to send some test G-codes to the CNC 3018 Pro

YOU'LL NEED

- **A CNC 3018 Pro kit**

There are lots of ways to create toolpaths for a CNC router to follow, and many CAD environments will have some kind of CAM software incorporated to create them. Most of the time, the CAM process takes a CAD model or drawing, creates a toolpath based on the geometry of the cutting tool, and then outputs a G-code file to be sent to the CNC machine. CAM software can be complex and so we wanted to find a package that anyone could use to create simple G-code files to get going with the CNC 3018 Pro, and suitable for us to test the machine with a variety of materials.

We found a brilliant piece of free and open-source software called KrabzCAM (**Figure 3**). KrabzCAM can run online by simply going to **hsmag.cc/krabzcam**, or you can download the zipped repository from the GitHub page (**hsmag.cc/krabzcamgit**), extract it, and run it offline. To do this, navigate to the **index.html** file to launch the software in your default web browser, running locally and not needing an internet connection (**Figure 2**).

KrabzCAM can import SVG or DXF files which many graphics packages are capable of creating. For our basic tests, we used the open-source Inkscape software to make our simple image paths.

Opening KrabzCAM, we click the 'Load SVG/DXF' button and import our SVG. Once our circle appears in the preview window, we can move it by pressing **A** to select it, and then **G** to grab and move. With the path selected, we can make a Pocket, Profile Inside, Profile Outside, and other operations relative to the

path. Pocket is where we would create a file with the tool, following a clearing pattern to remove all the material inside the path to a defined depth. Profiles are cuts into the material following either the inside or the outside of the path. For our first tests, we'll create an outside profile operation.

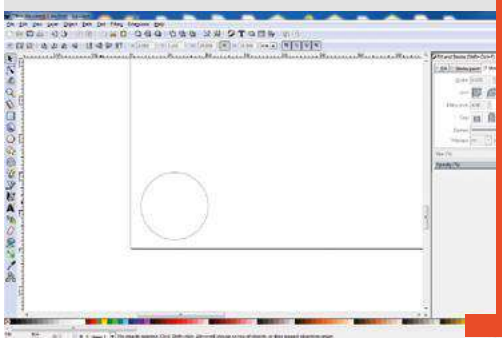
From the drop-down menu, select the Profile Outside option. In the dialog that appears, there are numerous settings we can adjust. For most of our test cuts, we used a 1.5mm diameter cutter that came with our machine. These carbide end mill burrs are commonly sold for PCB cutting and don't give the best finish on other materials. However, we have found that as a starter tool for soft materials, they are perfectly adequate.

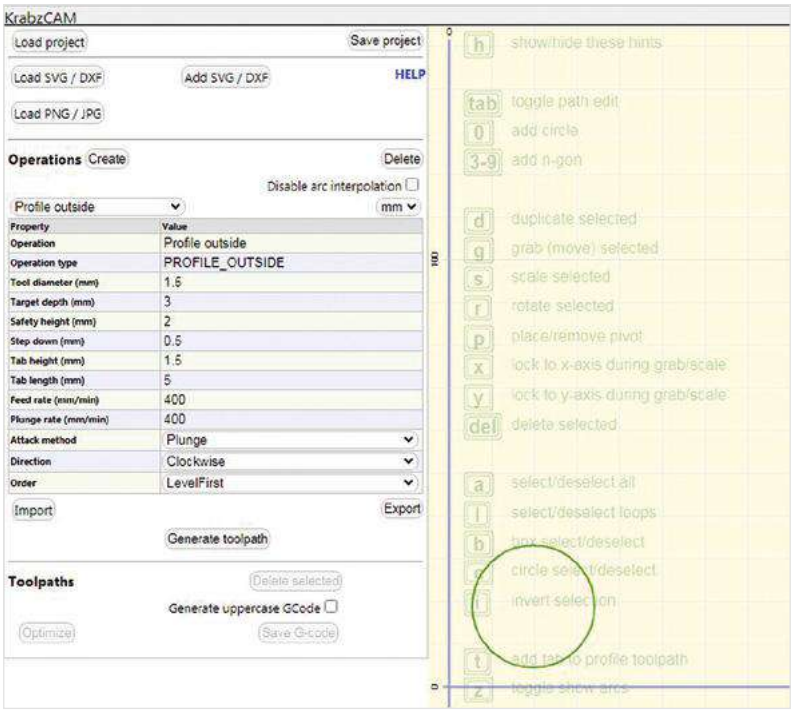
In KrabzCAM, set the tool diameter to 1.5. For the first test we used some 3.125mm plywood, aiming to cut all the way through. If we clamp the material to a waste board, that means that we can cut a little over the depth to 3.5mm, which will cut into the waste board but not cut into the machine bed. So let's set the 'target depth' to 3.5mm. Next, set the safety height to 5mm. The safety height is the height at which the Z axis moves to in order to travel over the work to its next cut position. With flat work, we could get away with 1 or 2 mm of safe height, but we tend to err on the side of caution and use 5mm.

TEST FILE

For our test cut file, we drew a circle in Inkscape using the circle tool. Using the fill and stroke tools available under the 'object' menu, we removed the fill from the circle. We set stroke thickness to 0.02mm, and sized the circle to 25mm. This makes the stroke very hard to see unless you click View > Display Mode > Outline.

We selected the circle object, and then clicked Path > Object to Path to ensure that our circle was a single contiguous path. Using the document properties, we then resized our document to be just a little larger than the circle. Finally, we saved the image as a standard Inkscape SVG.



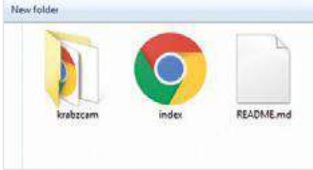


'Step down' is the depth of cut that the tool will take on each pass of a toolpath. Knowledge of the depth of cut for different materials comes with practice and wider reading. It's worth keeping notes of what settings you have used successfully (or not) to help you build experience. As this is a small machine, we went with quite a small depth of cut at 0.5mm. Of course, this means our job will do seven passes at 0.5mm increments to complete our 3.5mm deep cut. The next two items on the list refer to 'tabs'. Tabs are added to toolpaths to stop the work from becoming loose when it's cut through. We are going to add tabs after creating the toolpath, but →

Figure 3
Fabulous, free, and open-source software – KrabzCAM is simple and straightforward to create 2D toolpaths and G-code files

Figure 4
The settings we used for our test file, to be cut into 3.125mm thick plywood

Figure 2
KrabzCAM can be used via the website, or downloaded and run offline



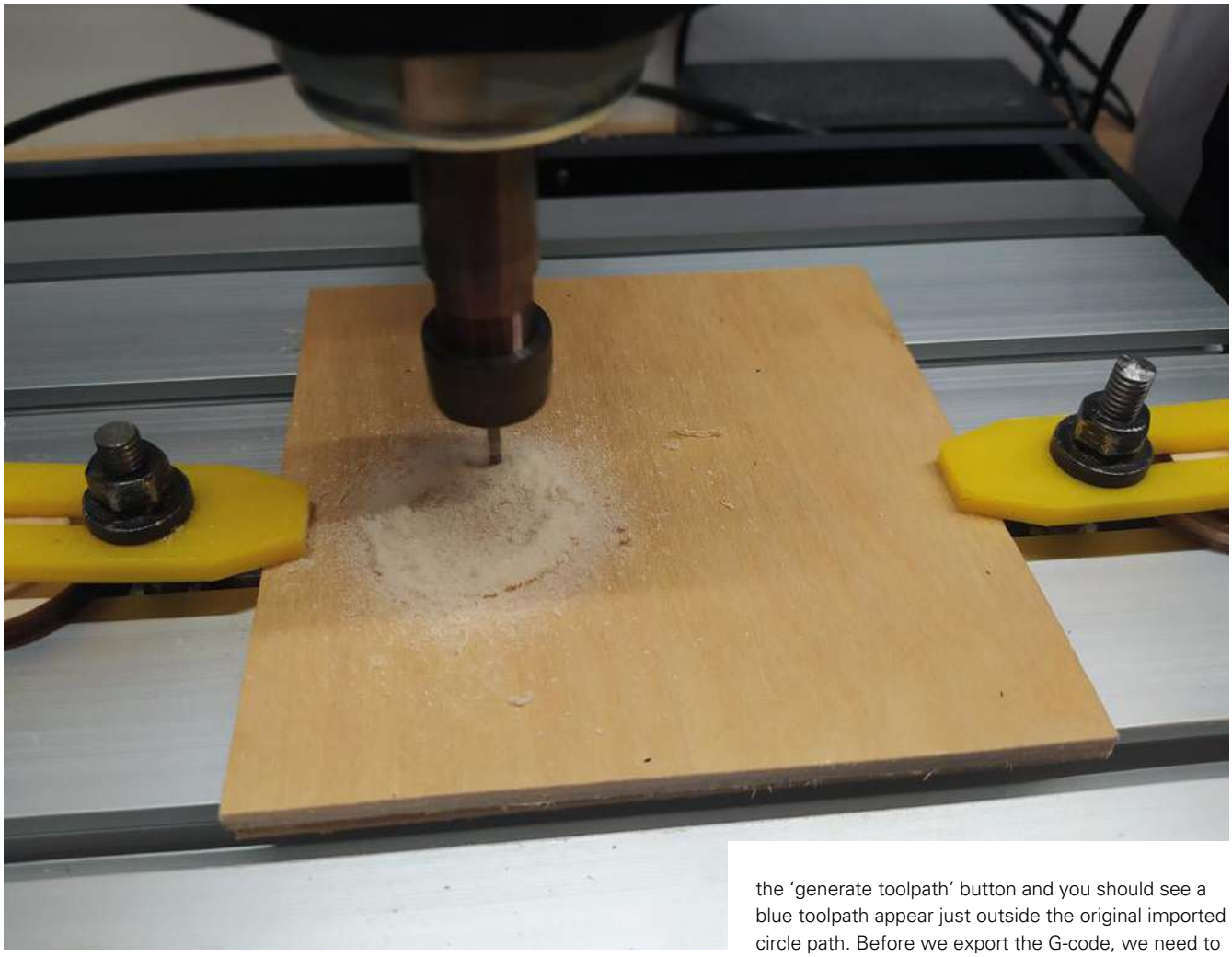


Figure 6
Our first proper cut under way!

Figure 5
Adding tabs to profile toolpaths means that the work won't become loose, and is held securely throughout the machining operation

let's define them now. Set the tab height to 2 mm and the tab length to 3 mm. Note that tabs are added from the bottom of the depth of cut and so if your depth of cut is beneath the workpiece, as ours is, this affects the actual height of the tab left on the workpiece. Cutting to a depth of 3.5 mm with 2 mm high tabs, in our 3.125 mm thick ply, will create tabs that will be 1.625 mm tall. 'Feedrate' is the speed at which the tool travels in the X and Y axis. Again, experience on different machines varies and we went with a conservative 400 mm/min. Plunge rate is the speed at which the tool travels down into the work material on the Z axis. We mentioned in the last issue that the CNC 3018 Pro inherently has some flex in the Z axis assembly, so it's a good idea to use slower plunge rates. Slower plunge rates enable the spindle to cut the material more readily, and will minimise the tool deflecting off the work and the Z axis flexing. We went with 200 mm/minute (**Figure 4**, overleaf).

Leave the other dialog items – 'attack method', 'direction', and 'order' – at their default settings. Click the 'generate toolpath' button and you should see a blue toolpath appear just outside the original imported circle path. Before we export the G-code, we need to add the tabs. Hover your cursor over the toolpath and press **T**. You should see a circle with a T inside that is attached to the toolpath but moves with your mouse. Move it to any position on the toolpath and left-click to add the tab (**Figure 5**). Repeat this for a second tab, roughly opposite the first. To export the G-code for our toolpath, first click the 'Generate Uppercase G-code' box, and then click the Save G-code button. Give your file a name, and then click Save, and your file will be downloaded to your **Downloads** folder.

WASTE NOT, WANT NOT

A good approach to setting up a CNC router is to add a waste board that covers the entire cutting bed, and then use small screws to attach stock to the bed. For these early tests, however, we went with simply clamping a small piece of waste board and the work stock down to the bed. The included clamps that come with the CNC 3018 Pro kit do work (as seen in **Figure 10**, overleaf), but the bolts provided are a bit long and can get in the way of the spindle somewhat. We tend to use these 3D-printed clamps with a short

piece of M6 threaded bar and some scrap materials to chock up the back of them. With either approach, there are a couple of key points about clamps. The first is that the clamping nut should be as close to the workpiece as it can be. In our slotted clamps, we always aim to have the nut and threaded bar at the 'nose' end of the slot. The second point is that you need to keep your clamping angle shallow. If you use the same materials stacked at the chocking end, you need only add a thin object to raise the angle a little. If you raise the clamping angle further, you are actually reducing the surface area of the clamping nose and it can grip less and also is more likely to distort the work stock. A final tip for clamping is that if you use the supplied clamps, make sure to put some scrap material under the bolt that presses against the machine bed to stop any damage from occurring.

> **Often, it's useful to have a ruler on hand so you can** quickly estimate if your job will fit

Turn on the CNC 3018 Pro and let's open the GRBL Control Candle 1.1.7 software supplied with the machine. Our version of Candle didn't need any installation, and would run quite happily from the supplied USB pen drive. You might need to install the supplied CH431SER driver, although our machine had it already installed. Once open, Candle automatically detected, and connected to, our CNC 3018 Pro.

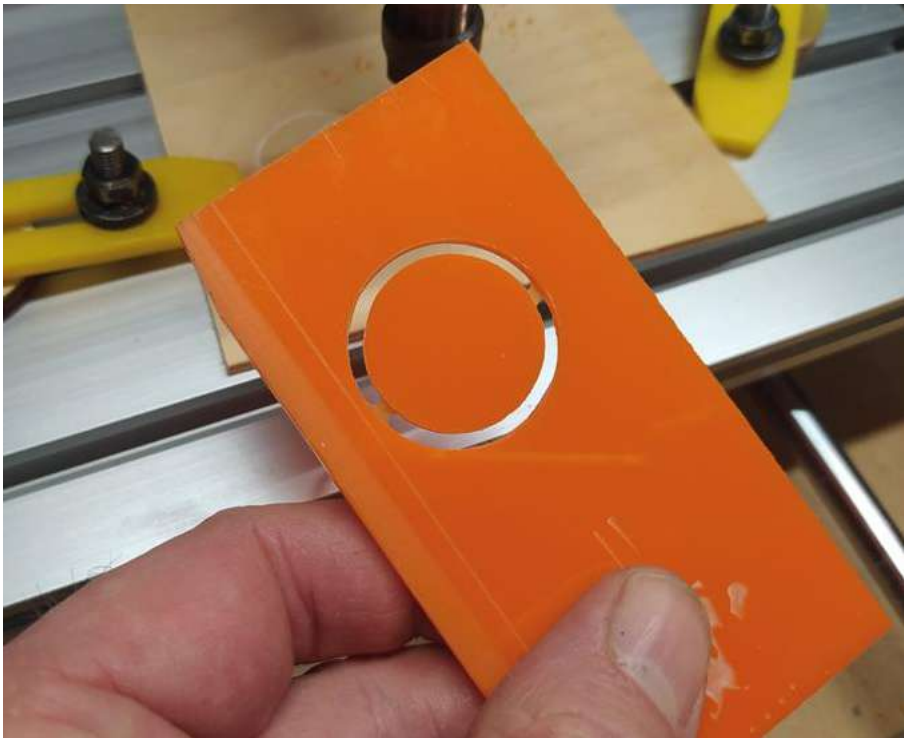


Figure 8 We created a new file to test the machine cutting Perspex

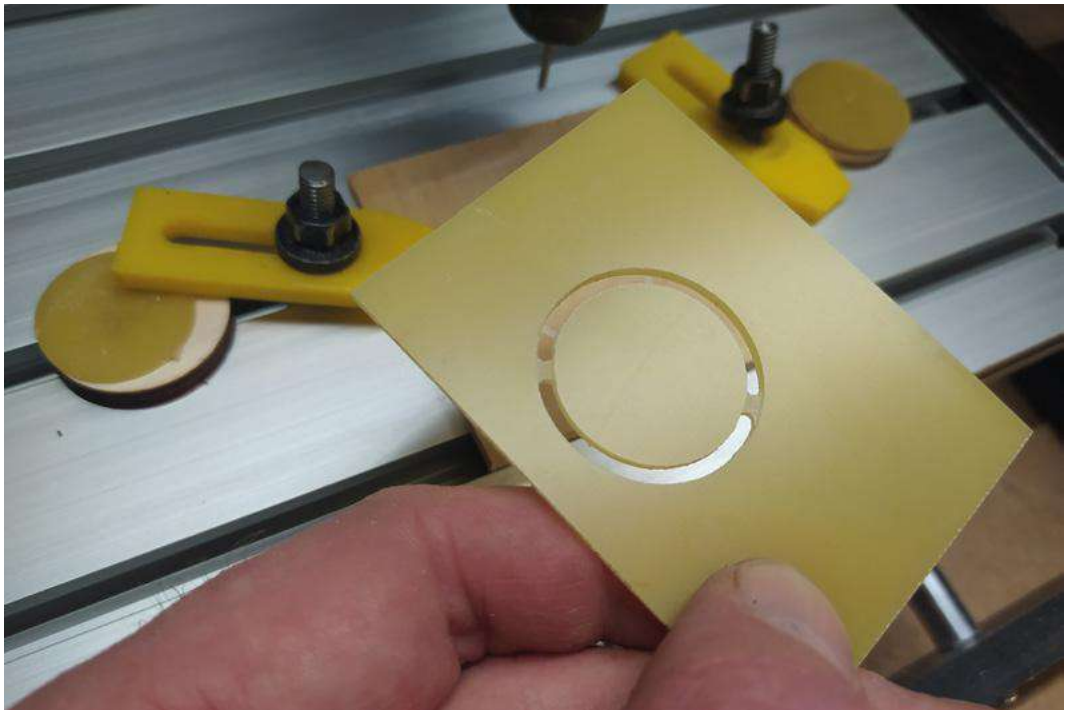
Figure 7 We were pleased to see excellent dimensional accuracy in our first cut part

The first test is to check our machine is correctly connected and configured. Make sure that the 'step' amount is set to something small like 1 mm and then, having checked your spindle isn't at the end of travel on any axis, use the jog arrows to create some small movements. Next, it's a good idea to click the 'open' button and load our G-code file. Loading the G-code file is useful, as it is visualised in the main window and, in the lower left-hand corner, the overall size of the job is stated (**Figure 1**). This isn't just the size of our 25 mm circle – it includes the area from the start point of the document, and also the diameter of the tool. This is handy, as you can ensure that you have this space available on the stock without running into any of your clamp arrangement or the edge of the stock. Often, it's useful to have a ruler on hand so you can quickly estimate if your job will fit.

You can then start to move your X and Y axes towards a good starting point on your workpiece. Once happy, you can click the 'zero x and y' button, second from the right on the top row of button →

QUICK TIP

We wrote about setting up toolpaths with FreeCAD using an earlier version of FreeCAD in issue 25. Watch out for another CNC/CAM article using the massively redeveloped FreeCAD path workbench in the future.

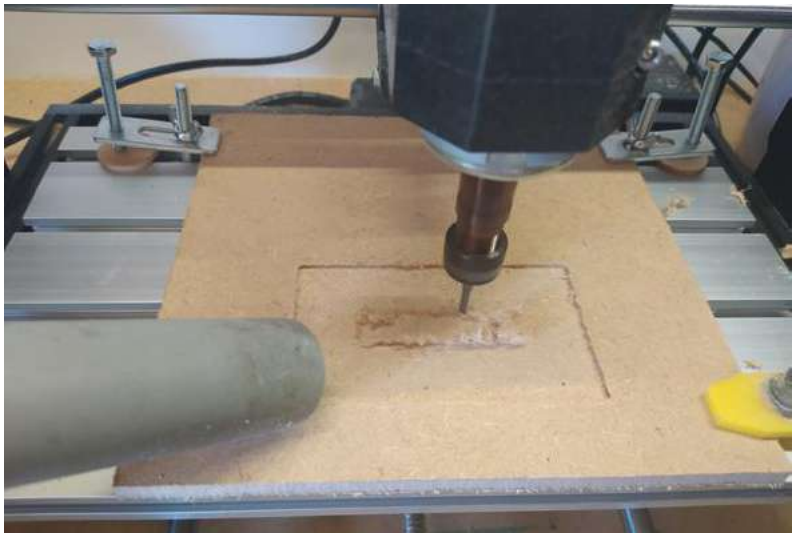


QUICK TIP

Remember to keep safe, tie back long hair, stow loose sleeves, and put on your eye protection!

Figure 9
Next, we repeated our 25mm circle test with a file created to cut G10 fibreglass sheeting

Figure 10
Using a different tool and using a 'Pocket' operation to remove material from a piece of MDF



icons. Jog the Z axis up a little to give yourself some room, and then fit the correct 1.5mm tool into the collet. Note that the collet inside the nut should be fitted into the nut before any tool is inserted. This is important when changing the collet size, which you may do if you buy some additional sizes. Ensure that the collet nut and the tool are secure using the supplied spanners.

Next, jog the Z axis down until the tool just touches the surface of the work. For this test piece, simply doing this by eye is OK, but for finer, more accurate work, we might use a feeler gauge to accurately position the Z axis. Using a feeler gauge of a known thickness (0.04mm, for example), we lower the Z axis in small increments until we can feel some resistance when moving the feeler gauge between the tool and the stock. Removing the feeler gauge, we can then set the travel to 0.04mm to bring it to the zero point exactly. Whether by eye or by feeler gauge, once in position, press the 'zero Z axis' button.

We are ready to start the job! Click the Send button and the job should begin (**Figure 6**). We found that the 1.5mm tool does an OK job on plywood, but leaves a slightly ragged edge on the top surface. Often, at the end of a job, we will jog the tool head away from the work and do a quick couple of passes with some sandpaper to remove the ragged edge. Whilst there are better tools to cut plywood, the supplied cutters are certainly adequate to get started with.

We used some side cutters to snip the tabs and remove our 25mm disc. We were happy to note that, with a pair of zeroed digital vernier callipers, our 25mm circle design measured 24.98mm (**Figure 7**). For a budget machine, that is a great level of accuracy.

To test a variety of materials for this article, we repeated the above process creating different G-code files for our same circle and 1.5mm tool, but adjusted the feeds and speed settings. First, we tried 3mm Perspex. We used a 400mm/min feedrate, with a step down of 0.5mm, and a plunge rate of 200mm/min. We turned down the spindle speed using 'M3 S800' most of the time with 3mm or less tools in the CNC 3018 Pro: we will want the spindle at full power, but we know that sometimes with Perspex,

a fast spinning tool can create heat and begin to melt the Perspex. Our experiment worked well, and again produced an accurate 25 mm disc (**Figure 8**). We could see some tiny anomalies where the Z axis plunges had occurred, so again we might experiment with slowing the plunge rate, or indeed experiment with ramping the tool into the work, slowly lowering the tool as it travels along the toolpath, creating a wedged shape/ramp into the workpiece.

PICK YOUR MATERIAL

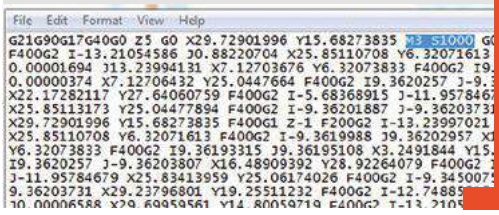
We repeated the process with some G10 fibreglass sheeting that was 1.5 mm thick. For this, we slowed the feedrate down to 300 mm/min, kept the step down at 0.5 mm, and the plunge rate at 200 mm/min. This is where the 1.5 mm tool shines, as this is the material the tool is designed to cut. It created a very clean cut in the fibreglass, with no need to clean up any surfaces or edges (**Figure 9**).

We tried something different with some MDF board. Swapping the tool for a 4 mm diameter, 2-flute end mill, we set up a pocket cut in KrabzCAM. Pocketing is where you clear the material out of the inside of a path to a specified depth. We drew a 100 mm by 60 mm rectangle in Inkscape and imported it to KrabzCAM. Selecting 'pocket', we then set the tool diameter to 4 mm, and set the target depth of cut to 2 mm. We set the step down to 0.7 mm so it would complete to 2 mm in three passes. The 'step over' is the percentage amount of the tool diameter by which the tool overlaps each toolpath. We left the step over and got good results, at 40 percent. We left the rest of the settings to their defaults and set the Z safe height to 5, feedrate to 400 mm/min, and plunge rate to 200 mm/min (**Figure 10**). As we weren't going to cut all the way through, we didn't need to add a waste board underneath our MDF. Running this job takes longer, as there is considerably more cutting taking place. We were pleased to see that it cut an accurate pocket with a very good surface finish in the MDF.

Creating a pocket like this is a useful technique that people use to create very level and accurate surfaces on their CNC machines. If, for example, we want to do some engraving where we only want to cut 0.1 mm deep into a workpiece, it's important that the work is held accurately perpendicular to the spindle, as any inaccuracy in an uneven bed can mean the work isn't being cut at all or cut too deeply. A pocket cut into the bed material creates a very accurate mount point for small engraving work. This will be useful next issue when we look at some engraving techniques, as well as some tests on tougher materials! ◻

HACKING G-CODE

Some CNC routers have their spindles set manually, not controlled by the G-code and, as such, KrabzCAM doesn't set the spindle speed in the files it produces. The CNC 3018 Pro control board does have the spindle controlled via G-code, and it's a simple edit to add a spindle command into the file we just created. Go to your **Downloads** folder and find the G-code file, and open it with a text editor. On our old Windows machine, we had Notepad installed. You should see a document with a heap of code inside. Don't worry particularly about what it all means, but you probably will find, as you explore CNC approaches, that you end up understanding quite a few G-codes. Looking at the beginning of the file, you will see the group of G-codes: G90, G40, and G21. Then you should see some code that reads 'G0 Z5'. This code is the first movement instruction of the file, and is instructing the machine to move from its current position to move the Z axis to the 5 mm position. So, if your machine's tool tip is touching the workpiece at the zero position, it would raise up 5 mm. After this, there should be an M3 code. The M3 code is actually a code to turn on the spindle, but it needs to be accompanied by a speed setting to let the machine know how fast to turn the spindle. The CNC 3018 Pro uses PWM to control the spindle, and the speed range is set using a proportional value between 0 and 1000. So, to turn on the spindle at full speed, we need to send the command 'M3 S1000'. However, we are going to move the command to a little later in the sequence to protect our workpiece. On our setup, if you send the G0 Z5 command and then the M3 S1000 command, the M3 S1000 command is run as soon as the Z axis starts to move. This can mean that the spindle can sometimes mark the workpiece at the zero point. To avoid this, remove the M3 command from the file and insert 'M3 S1000' after the next commands. The next G-codes after the Z5 command are another G0 command with some long X and Y co-ordinates. This command is moving the spindle to the start position of the job, the point at which the spindle will be plunged downwards to begin cutting. If we place the M3 S1000 command after this command, it means that the spindle will start to turn as the tool begins to move from the safe height position at the zero XY co-ordinate to the start point. Pasting the M3 S1000 command in the correct position in Notepad can be seen in the image below. Save the file once you have made the changes.



QUICK TIP

If you want to try different diameter shanked tools, as we have done, you will need to buy the correct diameter ER11 collets to hold them. They are available cheaply online.

Part 02

Build an arcade machine: Assemble your cabinet

Once your arcade cabinet kit arrives, it's time to put everything together

MAKER

K.G. Orphanides

K.G. is a writer, maker of odd games, and software preservation enthusiast. Their family fully supports the idea of an arcade machine in the living room.

@KGOrphanides

In this tutorial, we will assemble an arcade cabinet, fit controls, and mount a monitor. You should follow the video or written assembly instructions for the model you buy, but we'll go through the process so you know what to expect and how to handle the awkward bits.

Kits don't necessarily come with the screws and bolts you'll need to attach parts such as speakers, speaker grilles, and monitors, so check that you have all the hardware you'll need before you start.

Our cabinet is an Omniretro Bartop Arcade King with a stand (**magpi.cc/kingbartop**), made of 16 mm black melamine laminate, and we are using a 24-inch monitor.

01 Lay out your parts

MDF and melamine laminate are light, cheap, and sturdy when assembled, but can be susceptible to damage if dropped or pivoted hard on an edge or corner.

Make some space and put down towels to protect the cabinet parts and your floor from one another. If your unit consists of a separate bartop and stand, build them one at a time. Read or watch the manufacturer's instructions and make sure that you have all parts, fixings, and tools to hand before you start.

02 Preparation

Assembly varies from brand to brand. If access to the assembled cabinet is restrictive, you may have to fit your buttons and joystick to it before you put it together.

Similarly, attach speakers to the inside of the marquee bottom and speaker grilles to the outside before you assemble the cabinet. If you're working with laminate, mark up the screw positions through their holes with a paint pen and use a 3 mm bit to drill pilot holes.

If you've already decided on your marquee, control panel and bezel graphics, your life will be easier if you apply these to their acrylic sheets before assembly (we'll be looking at this in detail in a later tutorial).

> "Put down towels to protect the cabinet parts and your floor from one another"

03 Assembly

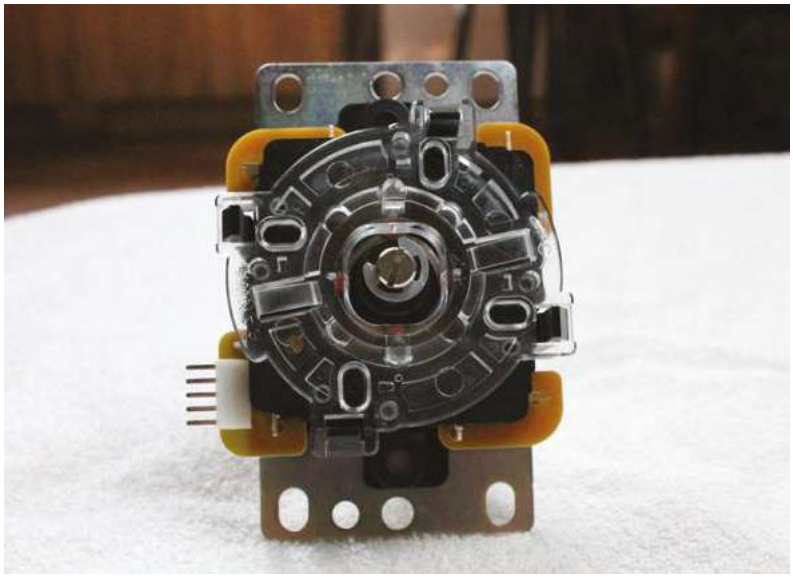
If you're comfortable with self-assembly furniture, an arcade cabinet shouldn't present too much trouble, but a second person can be helpful for fitting and moving awkward parts.

Ours has a control panel with a hinged access door beneath it, so we attached this hinge first →

You'll Need

- Screwdrivers, spanners, Allen keys, crimping tool
- Cordless drill
- Drill bit set. Screwdriver bits, drill bits, countersinks, tank cutters
- Additional bolts, screws, female spade connectors (to mount components)
- Dremel (recommended) and 3 mm drill
- Paint pen (silver if you have black laminate, black for MDF)
- Old towels or sheets to protect parts
- Foam cleanser and microfibre cloths (to clean your cabinet and acrylics)

TUTORIAL



▲ The underside of a Sanwa JLF-TP-8YT joystick. Note the e-clip securing the central shaft

then set the panel aside. We then attached the hinge for the bartop's rear access door and base, lined this part up with the cabinet's hood-like top, and bolted all of these parts to one side panel laid on top of them.

Lining bolts up with pre-drilled holes for this kind of build can be fiddly. If you have trouble, screw the bolts through the side panel until they're protruding, and use them to help find the correct positions.

04 The control panel

With one side now in place, slide in the control panel and bolt it to the same side as the other parts. Next, attach the marquee bottom that houses the speakers, which should already be mounted at this point.

With this model, we then close the latch on the rear access door and carefully flip the entire cabinet over onto the now-secure side panel. This is the best time to slide the marquee and screen acrylic panels into place. If you've not already applied graphics to them, leave their protective film on – it's easy to peel off later.

We now position the second side panel. We recommend again screwing in the bolts until they just protrude from the opposite side to help you lower the panel securely and accurately onto its pre-drilled holes.

05 Feel the power

Drill a hole at the back of your bartop and run the power bar's cable out through it to connect directly to a plug socket.

Some suppliers will wire a socket and bar for you, but note that international plug standards differ. Use a plug bar that can be surface-mounted inside of the cabinet.

While you're back there, cut a hole to accommodate a booted Ethernet cable or, more tidily, a screw-down Ethernet extension port. This will make Steam Link game streaming easier.

06 Extending your shaft

If your cabinet is over 16 mm thick, you'll want a longer than standard joystick shaft. Shafts are easy to swap, but watch out for parts dropping out.

Like most sticks, our Sanwa JLF-TP-8YT's shaft is held in place at the bottom by an e-clip. Hold the unit upside down, press on the bottom of the shaft with your thumb, and use a small flat-head screwdriver in your other hand to pull the clip off, using the slots in it. Pull the old shaft gently out from the top and push the new one in, carefully setting the pivot at the top and the spring and black plastic actuator at the bottom into place.

Use a thumbnail to depress the actuator and slide the e-clip back into place. You can also use pliers or your screwdriver to help push it on. For a demonstration, see this YouTube video on changing joystick shafts: **magpi.cc/joystickshafts**.

▲ To fit the VESA mount, place the cabinet face-down, then put the mounted monitor face-down on the front acrylic screen. Use a tape measure to help with positioning

Warning!
Mains electricity & power tools

Be careful when handling projects with mains electricity. Insulate your cables and disconnect power before touching them. Also, be careful when using power tools during this build.

magpi.cc/drillsafety
magpi.cc/electricalsafety

Cable tidy

Cable lacing is a cable management technique where a nylon cord is used to bind wires together. It can be used to create incredibly neat builds, like this Arcade Stick by Gordon Hollingworth, Raspberry Pi's Chief Product Officer.

Gordon learnt to cable-tidy this way as an apprentice for the MOD. "Tying the knot has to be done in a very specific way to avoid it looking untidy," he tells us, "basically a capital offence in the apprenticeship!" Gordon's cables have knots regularly at 1cm, which keeps them smart. "We learnt this way because when you put a box into a plane or tank with some equipment in it, the vibration will shake apart pretty much any connection in the first hour. So this was the way it was done when electronics was more about wires connecting things than PCBs."

You can buy nylon cord and learn more from RS Components (**magpi.cc/cablelacing**).

▲ Snap-in buttons are held in place by plastic clips. Connect your DuPont GPIO cables first to make internal wiring easier

> "There's room to slide the screw slots on most joystick mounting plates"

07 Installing your joystick

Two plastic dust washers come with Sanwa joysticks. Slide one onto the shaft before you mount the stick onto the underside of your control panel.

When mounting your joystick, position it, mark up the position of the top right screw-hole on the joystick's baseplate with a paint pen, and drill a pilot hole, being careful not to go all the way through.

Attach your joystick by that screw, make sure it's centred, and mark up the next hole or holes. There's room to slide the screw slots on most joystick mounting plates, so you've got a bit of wiggle room when it comes to the final fit.

Don't worry too much about the orientation of your joystick – position it where it won't get in the way of the rest of your wiring. These are nominally designated up, down, left, and right positions; you can reassign these through wiring and in software.

Finally, slide the second dust washer onto the shaft on the other side and screw the joystick's ball on.

08 World of buttons

Snap-in buttons are ideal for thick wooden cabinets – plastic clips hold them in position inside the holes drilled for them. If you have an acrylic cover for your control panel, the buttons will hold it in place.

It's a good idea to attach your spade connectors to DuPont GPIO jumper cables before installing them, but you'll have to connect the shared ground cable after they're in place. We wired GPIO to the right and shared ground to the left connector on each button, but it doesn't matter which goes where.

Where you have longer stretches between buttons, skip a connector on the ground chain to give yourself some extra cable to play with.

You can label the end of each GPIO cable for later ease of connection to Raspberry Pi, but they're not too hard to trace in most cabinets. →

Top Tip

Foot the bill

To help the cabinet stand on an uneven floor, you can fit four rubber feet to its underside.

▲ To make it easier to line up the sides of your cabinet with their pre-drilled recieving holes, partially screw in each bolt until a couple of millimetres protrude on the far side

> "If you find that you now can't reach or fit a part, don't panic"

▼ Before construction, lay out the parts of your bartop on some old towels to protect them

09 Bolt screen to VESA mount

Screen mounting can be fiddly. Most cabinets come with a baton-like wooden VESA mount that's designed to be screwed into place from the inside. Start by bolting your monitor to the mount. Unless you're working with a specialist cab designed for giant screens, you'll be using a 75 mm or 100 mm VESA mount. These usually take M4 bolts and have a depth of 10 mm. So if bolts aren't included, you'll need four, at a depth of 10 mm plus the depth of your mount, although you can get away with shorter if you countersink them.

10 Screw VESA mount to the cabinet

Place the bartop face-down on the ground, protecting it with a towel. Take the protective plastic off the acrylic on the inside of the cabinet. Clean the acrylic with a microfibre cloth and anti-static foam cleanser

Lay the monitor, attached to the cabinet's VESA mount, face-down on the acrylic inside your cabinet. On the interior sides of the cabinet, mark up the position of the holes in the brackets on each edge of the VESA mount. Remove the mount, drill pilot holes, then replace and screw down the display and its mount.

If your monitor has a front power button, you can use adhesive chair leg floor protectors as soft spacers to stop it from being pressed by the acrylic screen.

11 Don't panic

If you miss a stage in your build and find that you now can't reach or fit a part, don't panic. Speakers – and any other components in need of securing – can be attached internally using strong double-sided foam tape.

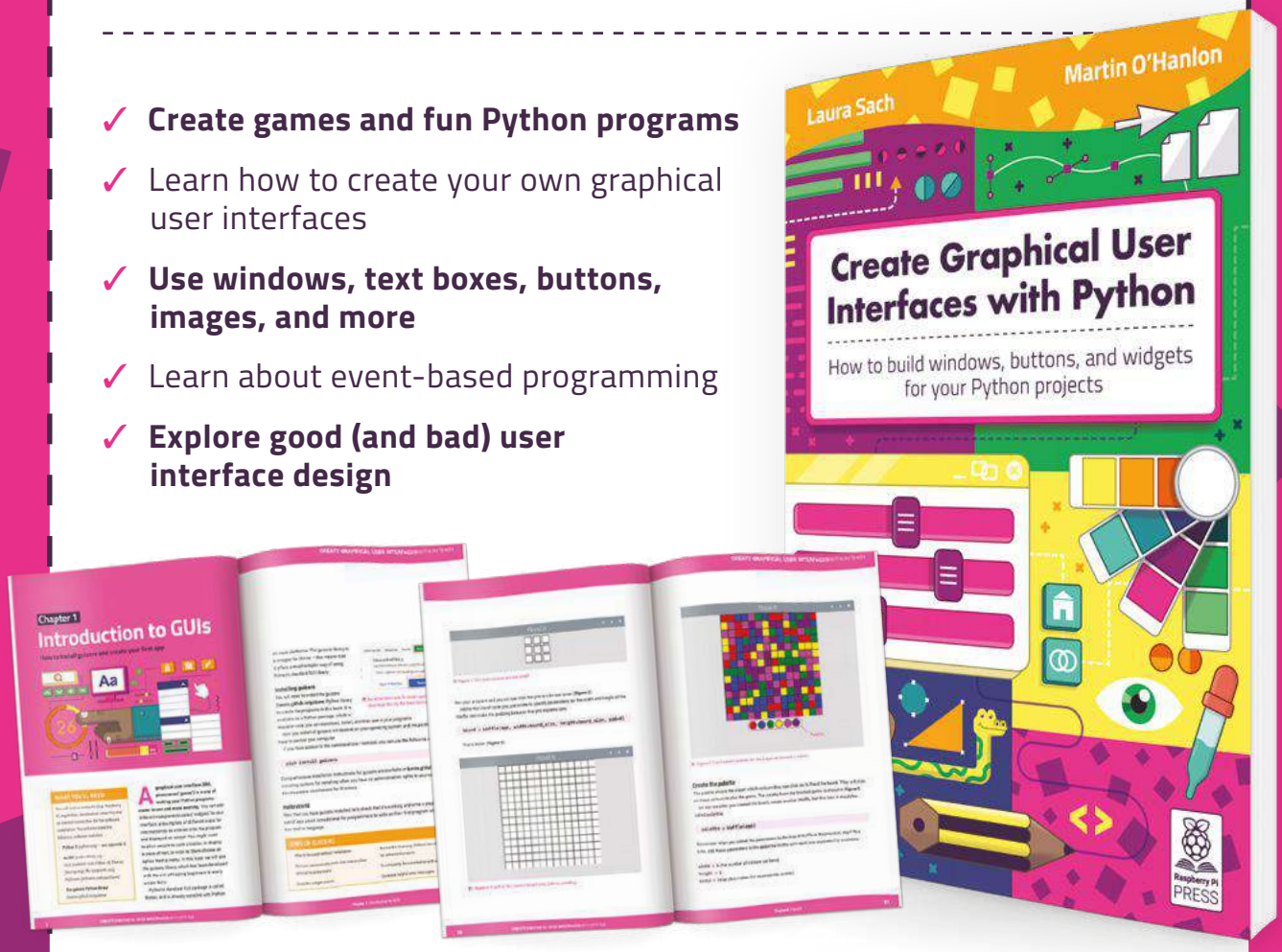
Most external parts can be drilled and fitted in situ. If you want to deal with decoration last, then you can sometimes pop out your acrylic panels or, better, remove one side and reattach it.

As you'll see from the photos, we have temporarily applied U-moulding to protect the edges of the cabinet. U-moulding is easy to remove and refit or replace, assuming you don't glue it down, but T-moulding is a little harder to remove cleanly.

We're now ready to connect Raspberry Pi. That will be covered in the next tutorial. ◻

Create Graphical User Interfaces with Python

- ✓ **Create games and fun Python programs**
- ✓ Learn how to create your own graphical user interfaces
- ✓ **Use windows, text boxes, buttons, images, and more**
- ✓ Learn about event-based programming
- ✓ **Explore good (and bad) user interface design**



Buy online: magpi.cc/pythongui

FreeCAD: getting started with assemblies

Make complex designs out of a collection of parts

Jo Hinchliffe

@concreted0g

Jo Hinchliffe is a constant tinkerer and is passionate about all things DIY. He loves designing and scratch-building both model and high-power rockets, and releases the designs and components as open-source. He also has a shed full of lathes and milling machines and CNC kit!

In the last issue, we used add-on workbenches. In this issue, we will use two other add-on workbenches, which we'll need to install. Open FreeCAD, go to Tools > Addon Manager, and click to install the A2plus workbench. Once installed, follow the prompt to restart FreeCAD, and then repeat the above task to install the Fasteners workbench.

Creating an assembly in CAD is useful for many reasons: it allows us to design more complex projects, to check that parts fit, and it allows us to use the same parts multiple times. It also can be used to check clearances and movements of dynamic systems and more. We are going to make and assemble an example of a pulley wheel mounted in some brackets, sat on top of a base, and held together with M6 nuts and bolts. As we are focusing on the assembly, we aren't going to step through how we made every separate component, but we have posted the parts used in this project online, so you can just have a go at assembling the pulley block using those parts. This isn't supposed to be a particularly well-designed pulley block; it would need some bushings and bearings in real life, as well as a proper spindle, but as an example, it's a good project to play with.

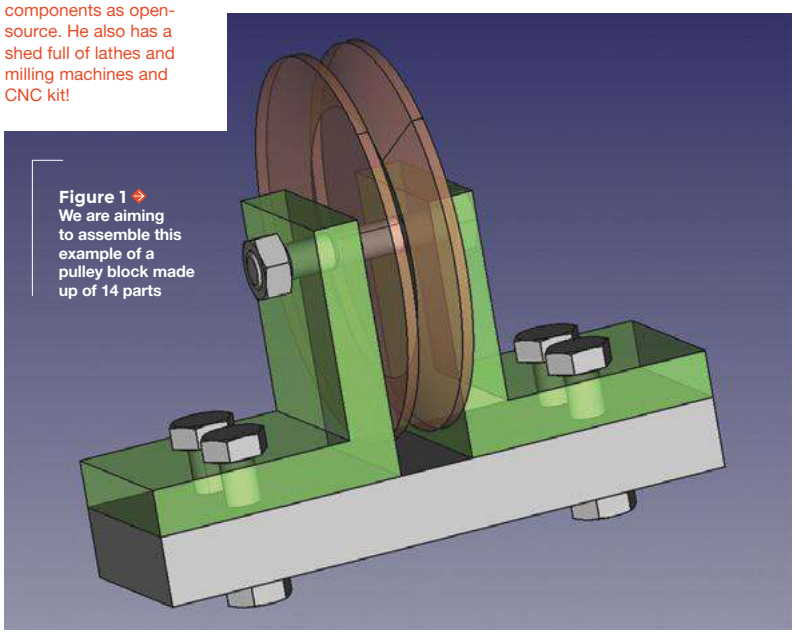


Figure 1 We are aiming to assemble this example of a pulley block made up of 14 parts

When using Part Design to create a body, that body is a single continuous solid object. The classic example is a nut and a bolt – which would require two bodies: one for the nut and one for the bolt – because although the items can be assembled, they are both single solid items. You could make two separate bodies within a project and then painstakingly move them into position relative to each other, but this is long-winded, and also, if you make a change or move the bolt, suddenly the nut is in the wrong place. Using A2plus, we will make assemblies where components are constrained to the specific parts of other components that they are attached to. Beyond parts simply being lined up positionally, the assembly constraints describe the relationship of one part to another. For example, if something can turn around an axis, or if one component is always parallel to another.

> "This isn't supposed to be a particularly well-designed pulley block **... but as an example, it's a good project to play with**"

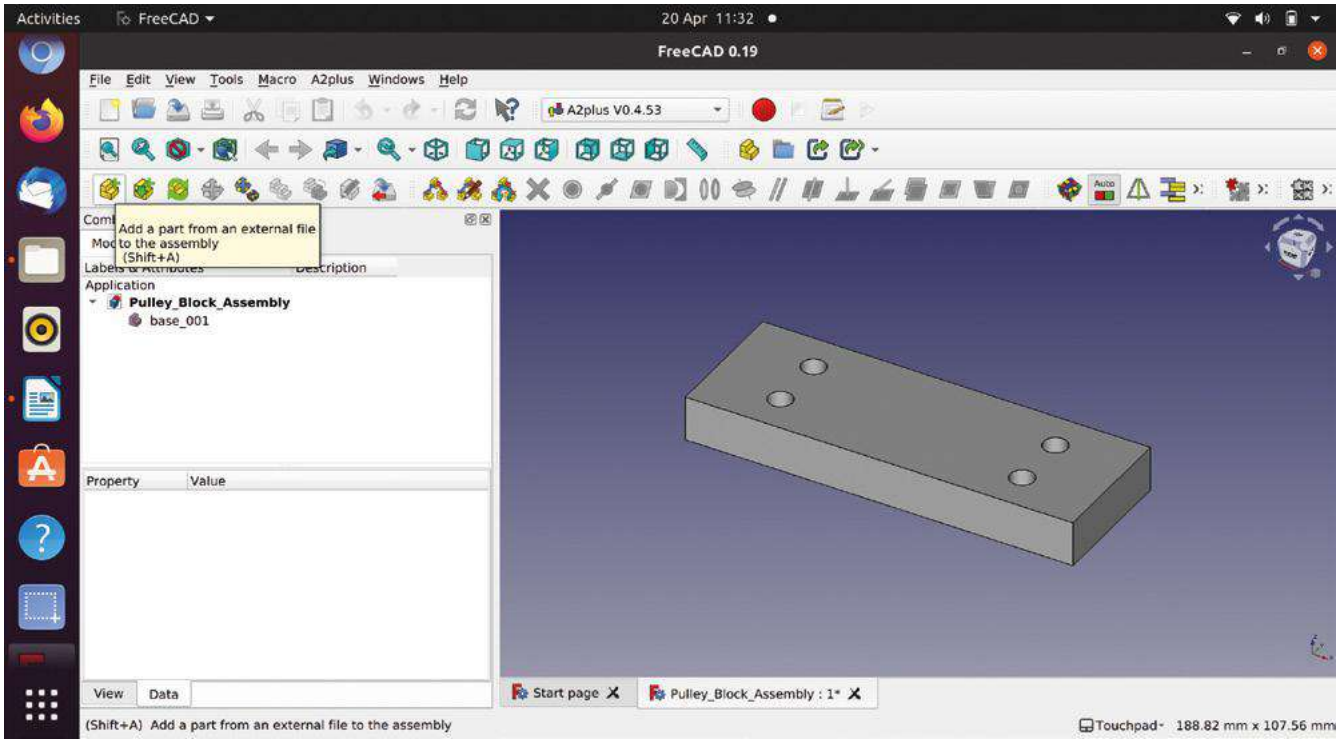


Figure 2
Importing our first part of the pulley block

We began by making our components using Part Design to create bodies for each part. We created a base, two L-brackets, some nuts and bolts, a simple V-pulley, and a longer nut and bolt to act as a shaft. You'll see that we are going to import these items into the A2plus workbench, and when we do, we could import each body either from an individual project, a project with just that one body in it, or we can import multiple bodies from the same project. Most of our files are single items, but we created one file containing two parts to allow us to explore how to import an individual part from a multipart project. This is useful for many reasons. We can imagine borrowing items from unrelated projects in our library.

We made a base which is a simple block and is the file 'base'. The base has holes to mount the L-brackets using M6 nuts and bolts. The sketch for the base is drawn in the XY plane and is centred around the 0,0 point – the bolt-holes are constrained positionally relative to that origin point. This means that if we change the size of the base, the bolt-holes will remain in the correct position, which is important because they define the L-bracket position and the width of the gap for the pulley wheel.

The L-brackets are again a simple part with holes that match the hole coordinates on the base and a hole on the upright to receive the spindle bolt. You can see the complete assembly in **Figure 1**.

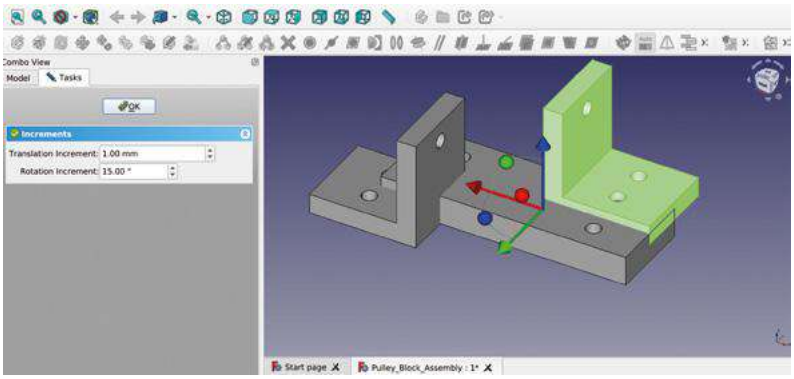


Figure 3
Importing the L-bracket file twice, we will position these using the constraint tools to attach them to the base

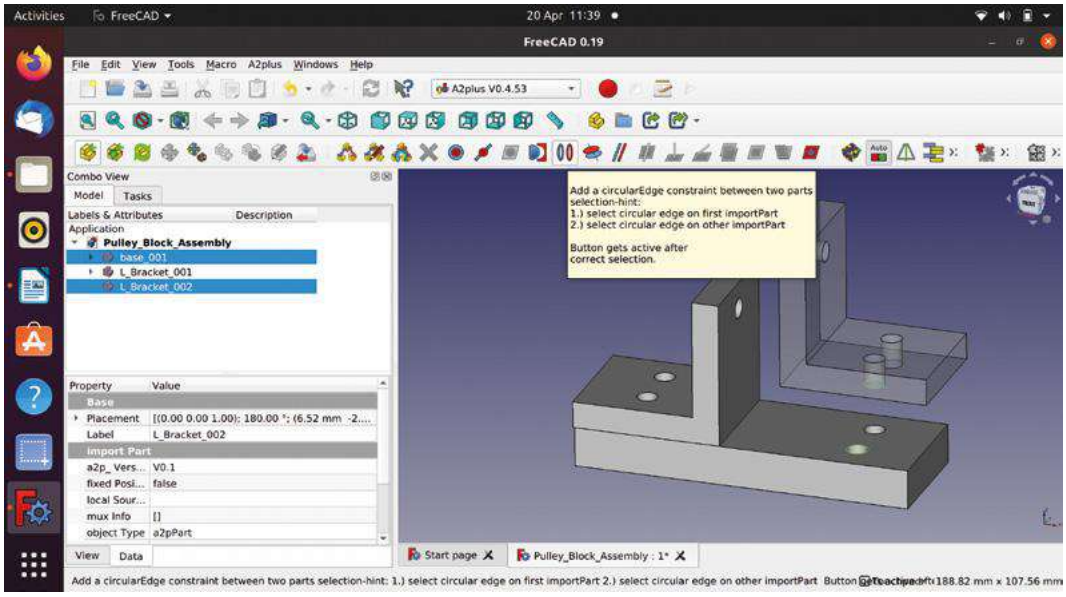
Download the files (from **hsmag.cc/issue43**) and create a new project in FreeCAD. For ease, save the new project in the same folder as all the component files. Move to the A2plus workbench and start importing components. On the left-hand side of the A2plus toolbar, you should see a yellow and green tool icon, which when you hover over reads 'Add a part from an external file to the assembly' (**Figure 2**). Click this and navigate to select the 'base' file in the folder. Select the file and click Open. In the preview window, you should now have a copy of the base object inserted (**Figure 2**). Let's repeat the task with the L-bracket file. The L-bracket will import overlapped with the base – you might notice that until you left-click, you can move it to any position in the preview area. If you need to move it again, double-click on the item in the file tree menu and you can →

YOU'LL NEED

A computer with FreeCAD 0.19

QUICK TIP

We used the Part Design workbench and Sketcher workbench tools to create the different parts for our assembly. We covered these approaches in the first two parts of this series, starting in issue 37.

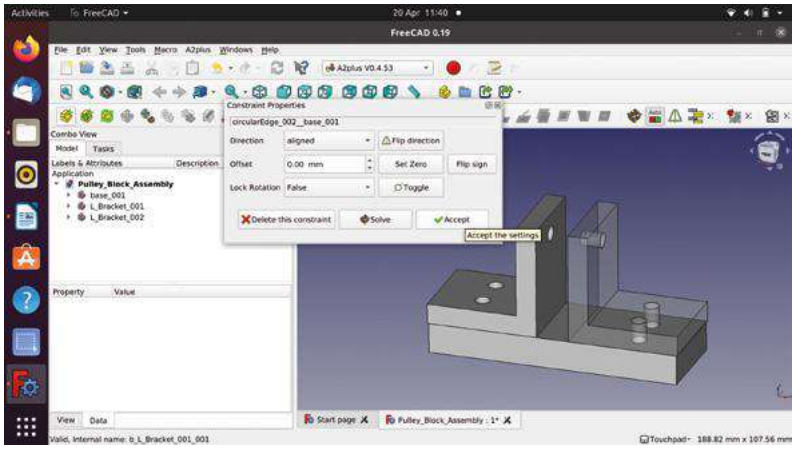


transform it incrementally using the dialog and the axis arrows. Move it to a position free of the base to make things easier to see. Repeat the process and import a second instance of the L-bracket file. The files in the file tree will automatically be given a number suffix, so we have **L_Bracket_001** and **L_Bracket_002**. As an example of one way of working, simply rotate the second bracket so that it faces the correct way relative to the base in terms of how it will sit when assembled (**Figure 3**).

Let's assemble the L-brackets to the base – to do this, select the bottom edge of one of the holes on the first L-bracket. This edge has a corresponding edge on the upper end of a hole in the base, and when in position, these two edges would sit together (**Figure 4**). Move the model so that you can see that particular hole edge, and then using the **CTRL** key, click to select it. You should now have two hole edges selected. When you select items in A2plus like this, any constraint types that could be applied to the

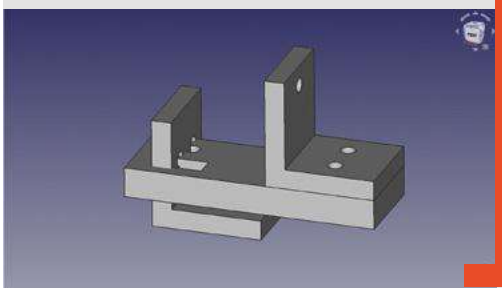
Figure 4
Setting some parts as transparent enables us to show which hole circles we selected to constrain the L-bracket to the base

Figure 5
When applying constraints, you will see a dialog box with options



KEEPING BACKUPS

If you hang out in CAD circles for long enough, regardless of what CAD environment you use, you may come across a term for a problem. The 'topological naming problem', or 'toponaming' for short, is a common issue we can easily demonstrate in our project. Objects and parts – for example, the base of our pulley block – are all made out of edges and faces. These edges and faces are automatically given an annotation. Sometimes you see these names when you select things like 'edge 12' or 'face 3'. When we apply constraints in the A2plus workbench, we are essentially telling FreeCAD to 'attach face X to face Y' or 'make edge 2 parallel to edge 5', etc. When we edited our base part to make the brackets fit, we only changed a dimension, and, as such, none of the names of the features changed. As an experiment, edit the base and create a small rectangular pocket over one of the holes – to any depth – and then save the file. As we did before, update the parts in the assembly; when you do so, you'll see that the model is now a little broken, as in the image below. This is because the names of the edges and faces have changed, but the constraints don't know this. It's definitely worth having a backup of a saved assembly before making any major edits to parts within it. You might find that sometimes these issues are easily resolved by removing constraints and reapplying them.



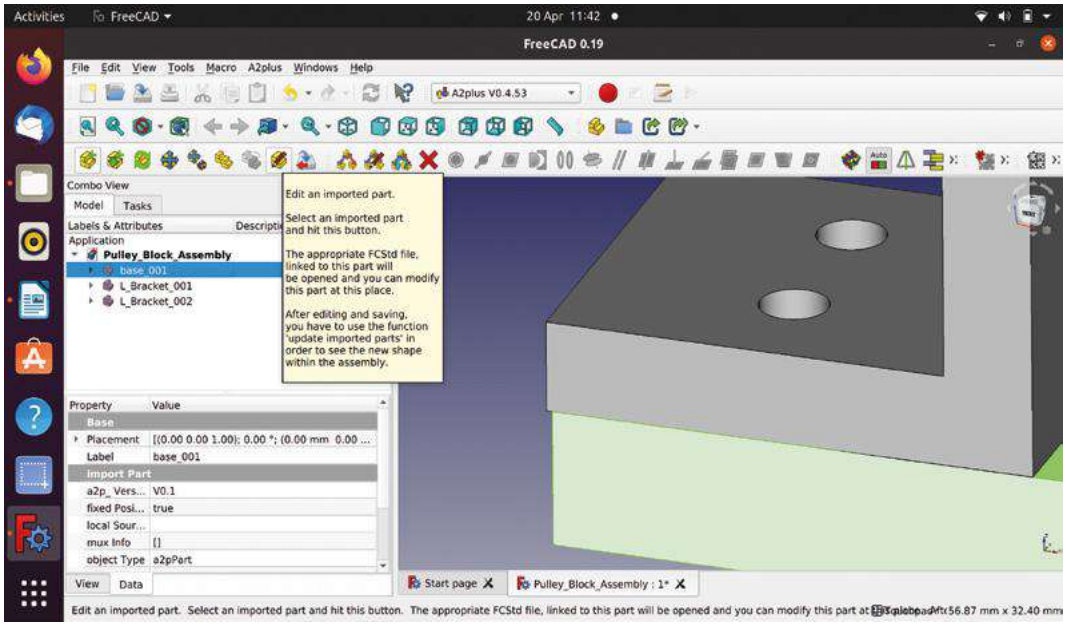
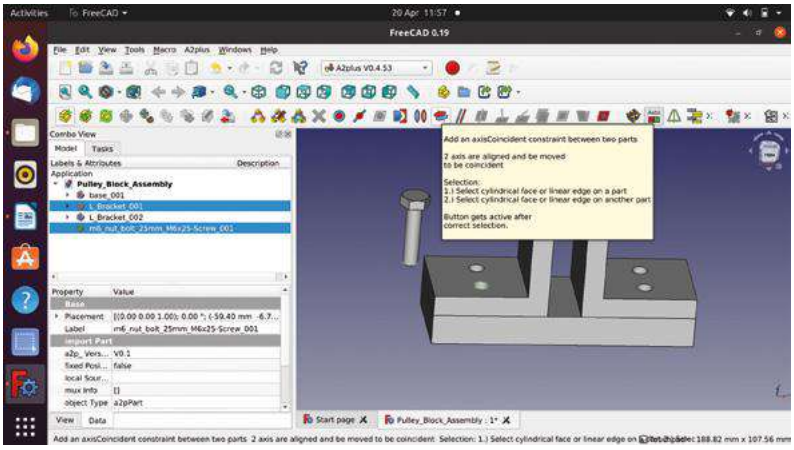


Figure 6
We can see the base doesn't quite fit; the A2plus workbench makes it easy to edit imported parts

Figure 7
Selecting a circle edge on the bolt and a circle edge on the hole to align the bolt and hole axis

items automatically become available on the toolbar. You should now be able to click the 'Add circular edge constraint' tool icon (**Figure 4**). You will see that the L-bracket moves, hopefully into a correct position. You will also see a dialog box with some options showing the constraints you can adjust. We shouldn't need to adjust the constraint for this one, but it's worth noting the 'Flip' button, which is commonly used if the constrained items end up in an undesired position – often clicking the Flip button repeatedly will toggle through the optional placements until you find the one you require. Click Accept to confirm the constraint (**Figure 5**).



DOUBLE UP

Repeat the process for the second L-bracket to end up with the two L-brackets facing each other with the gap in between for the pulley wheel. Noting the error on the size of the base, let's use that to work through how to edit a part from the A2plus workbench.

Highlight the base in the file tree and click the 'Edit an imported part' tool icon (**Figure 6**). Clicking this should open the base part in a separate project, and the file tree of the part will appear. Even though you are still on the A2plus workbench, you can click the drop-down menu to select the primary rectangular sketch, which is the root of the base object. Double-click this to open the sketch in the Sketcher workbench, and then change the horizontal dimensional constraint to 101 mm. Close the sketch and then save the base part project by clicking the Save button. Now close that project tab. Notice that the base in our partly assembled project hasn't changed. To update the parts, click the 'Update parts, which have been imported to the assembly' tool icon (a yellow part icon with two curved green arrows over it). Once clicked, you should see the base now fits, perfectly matching the L-bracket ends. A point to consider again is that the holes in the base file are constrained to the centre line of the base. This means as the base dimensions are expanded, the holes stay the same distance apart. If we had constrained the holes relative to the edges of the base, then our update to the horizontal length would have moved the holes, and the brackets would still overhang. As you develop parts destined for assemblies, these factors become more important to consider.

Using either the project with the M6 nuts and bolts you just made or, if you didn't, the file we supplied in the download, let's add the four nuts and bolts to connect the brackets to the base. Just as an experiment, try and import the nuts and bolts project using the same technique we did for the base and →

QUICK TIP

At the risk of sounding obvious, you can use a component part numerous times in an assembly, so don't model things more than you need to!

QUICK TIP

Similar to Sketcher and other parts of FreeCAD, there are numerous ways to achieve similar results using different assembly constraints.

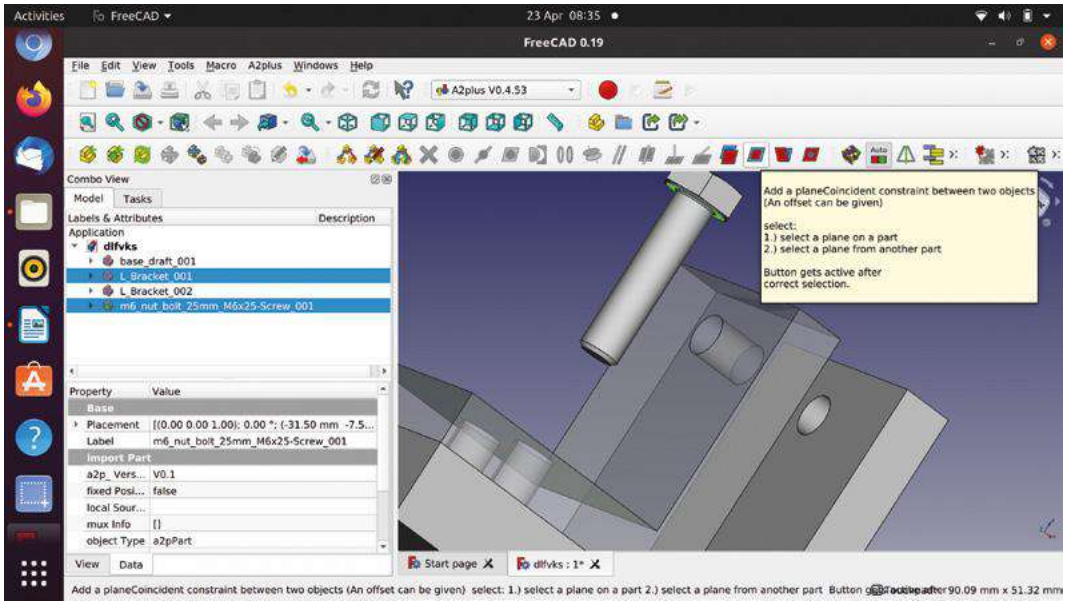
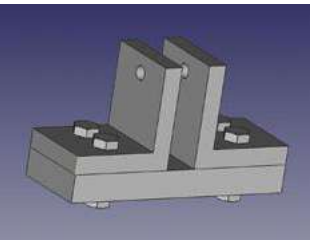


Figure 8 Selecting the top face of the bracket and a lower face of the bolt-head to apply a planeCoincident constraint

Figure 10 All the nuts and bolts added to the assembly



the brackets. You'll note that even though there are two objects, it will import, but the nut and the bolt appear as a single item in the file tree and cannot be moved independently. The correct way to do this is to use the 'Add a single shape out of an external file' tool icon. Having deleted our incorrectly imported nut and bolt, click this and then select the file again; this time, a dialog asks us to specify which part to import from within the file. Select the nut or bolt and repeat this for the other part. Once you have the first nut and bolt in place, let's constrain this one before tackling the others.

Let's constrain the bolt section first. Select one of the circle edges at the opposite end of the bolt from the head. Using the **CTRL** key, select one of the upper edges of one of the holes on the bracket. We want these two circles to align on an axis, but we don't want to constrain the edges together as the bolt end circle obviously needs to be out of the other end of the hole. Clicking the 'Add an axisCoincident constraint between two parts' achieves this – the bolt should now align to the hole (**Figure 7**, overleaf). To make the bolt-head sit flush to the surface of the bracket and the bolt to sit all the way through the hole, let's add another constraint. This time, click to select the upper face of the bracket and then select the lower face of the bolt-head. These two faces can then be brought together by clicking the 'Add a planeCoincident constraint between two objects' tool (**Figure 8**). Now the bolt should be correctly in position. Click Accept in the dialog box to finish the constraint. Moving to the nut, if we move the nut closer to the bolt end under the base, we can then select the edge at the base of the bolt before the slight chamfer and the lower similar edge just inside the nut (**Figure 9**). Again, apply a 'circularEdge' constraint to fit the nut to the bolt.

> "It's a great workbench to play with and experiment **with how the different constraints work**"

Now that our nut and bolt are constrained, we want to add the other nuts and bolts. Again, as an example for learning, you might think – based on other workbenches – that you can copy and paste the nut and bolt in the file tree system. Of course

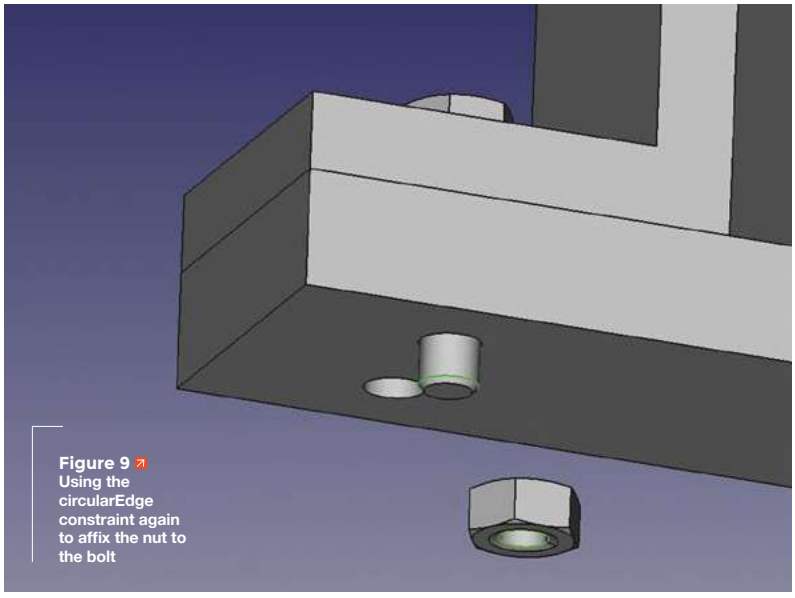


Figure 9 Using the circularEdge constraint again to affix the nut to the bolt

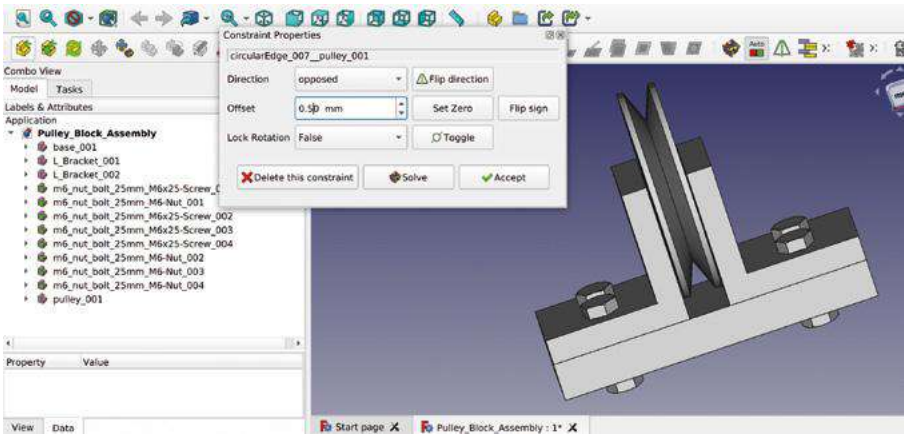


Figure 11
Adding the pulley, and positioning it with an offset within the constraint, to create the clearance gap with the L-brackets

you can, but now that we have added constraints to our first nut and bolt, if we copy and paste, that will copy all the constraints applied to that part. Of course, we could delete these constraints and reapply them in the other positions, but it's easier to either import multiple copies or to copy and paste before adding constraints to an object. However you choose, create three more nuts and bolts and constrain them into position in the assembly (**Figure 10**).

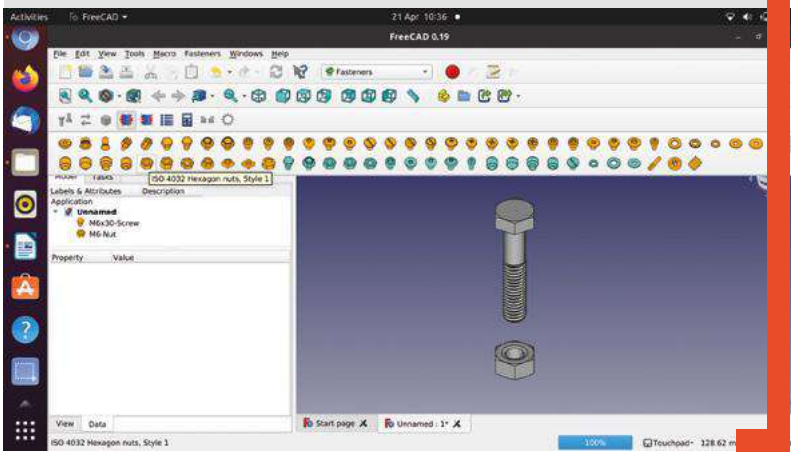
PULLING TOGETHER

Let's add the pulley wheel next. Import the pulley wheel, but let's use constraints to position it rather than double-clicking to use the transform tools. Select one of the edges of the hole through the pulley and then click one of the holes on the inside of the L-brackets in the gap for the pulley wheel. Even if the pulley wheel is at 90 degrees to the brackets, clicking the 'Add a circularEdge constraint' should bring it into the correct orientation. You may need to flip the direction in the constraint dialog to get the pulley largely between the brackets. Before you close the constraint dialog, note that the pulley is 12 mm wide and the gap between the brackets is 13 mm – this is because one would probably want the pulley wheel to have some clearance (**Figure 11**). Currently, the pulley is positioned touching the bracket of which we selected the hole edge. Setting the 'offset' to 0.5 mm in the constraint dialog moves the pulley 0.5 mm off the bracket and positions it equally with clearance between the two brackets. Finally, there is a 'Lock Rotation' value in the constraint dialog. Setting this as 'false' means that our pulley wheel part can be rotated around this circular constraint, emulating its real-world movement. If you wanted to lock it at a certain point in rotation, set the part in position and then toggle this value to 'true'.

To finish our assembly, import the spindle bolt file and constrain it through the centre of the brackets and the pulley, then fit the nut to the other end. We are sure by this point that you can work out how to constrain this one into position without instruction. We hope that you've found this introduction to assemblies useful, and after you have put this together, it's a great workbench to play with and experiment with how the different constraints work. The simple assembly we've created is a good starting point and a foundation for making assemblies that move and can be animated, and much more.

JOIN TOGETHER

We are going to make some nuts and bolts for our assembly. For this, we will use the Fasteners workbench we installed. Create a new project and move to the Fasteners workbench. The Fasteners workbench makes it easy to model metric and imperial nuts and bolts and screws – it even has some rarer fasteners like PEM insertion nuts, PCB standoffs, and more. We are going to use four M6 nuts and bolts that are 25 mm long to attach the L-brackets to the baseplate; however, we only need to model one and then import four copies to the A2plus assembly project. To create our bolt, first click the yellow icon that looks like a hexagonal-headed bolt, which when you hover over it says 'ISO 4014 Hex Head Bolt'. M6 is the default diameter for this bolt type on the Fasteners workbench – it should create a 30 mm M6 bolt model. You'll notice that the model doesn't have any thread modelled onto it. This is common across most CAD platforms when modelling threaded fasteners because the thread is a complex geometry. If you have lots of fasteners, it can increase the project complexity and speed of loading or recomputing. Most of the time in CAD, we don't need to model the thread; however, you can simply add the thread to this model by highlighting the item in the file tree and, in the dialog box, setting the 'thread' property to 'true'. Depending on your computer, this may take some time to generate the thread on the model. Turn the threads back to 'false' and shorten our bolt to 25 mm by selecting 'custom' in the 'length' property in the dialog box and typing in 25 mm. Let's create a nut for our bolt in the same project file. Depending on your screen, you might need to expand the toolbar to select the yellow icon that looks like a hexagonal nut labelled 'ISO 4032 Hexagon nuts, Style 1'. Again, the Fasteners workbench will automatically create a nut, but it will be positioned on the origin point in the same place as the bolt-head. To move this, double-click the nut in the file tree and then use the arrows in the preview window to move it down and away from the bolt. Making sure you have turned off the thread setting for each item, save this project ready for use in our assembly, or use the one we have provided.



TUTORIAL

Threadbare metal repair

Learn how to reline and rescue threaded holes with a thread repair kit

Dr Andrew Lewis

Dr Andrew Lewis is a specialist fabricator and maker, and is the owner of the Andrew Lewis Workshop.

Above
The bolts that hold the jaws on a vice often get damaged, because they're blind holes tapped into cast metal. If bolts break or rust, they also damage the threads and can be difficult to remove from the holes. Sometimes drilling out the broken bolt is the only solution

Right
Thread kits can be expensive to buy initially, but once you have the special fitting tools, you can sometimes save money by just buying inserts and taps for other sizes

Have you ever experienced that horrible moment as you're tightening up a bolt, and the spanner suddenly lurches forward and then goes slack? It's a sure sign that the threads have given way on whatever you're tightening, and it's probably time to take a breather, go and put the kettle on, and think seriously about buying a torque wrench. For some projects, a stripped or damaged thread isn't a huge problem. You can just redrill and tap the hole with a slightly larger thread, or exchange a damaged nut or bolt for a new one. But if the stripped thread is part of a large casting or milled object like a car engine or a machine vice, you could find yourself looking for a very expensive replacement part.

RIGHTY TIGHTY… SOMETIMES

Thread repair kits have been around for a long time, but it's surprising how few people know about them, and how they work. A thread repair kit (or Heli-Coil as it is often known) is a metal insert with threads on the outer and inner sides. The idea behind a Heli-Coil is that you can redrill and retap a hole, then screw in a threaded insert with an internal thread the same size as the original damaged thread. This is very convenient if you need to use a specific size of bolt that passes through several different parts, and simply using a bigger bolt wouldn't be possible.

The repair process is straightforward, but you will need to make sure that you have all of the necessary tools to insert and remove the bushings before you start. Most kits include a fitting tool, a drill, and a tap sized to match the Heli-Coil you're using. Buying all of these parts can really push up the cost of a repair, but once you have the tools, you can reuse them for multiple repairs.

MATCHING INSERT

You'll need a correctly sized threaded insert to repair a damaged thread, so if your damaged tap is M8 thread, you'll need an M8 insert. This doesn't usually cause any problems, but some older threads like Whitworth (which are often found on old vice jaws) aren't as easily available as modern metric sizes.

DRILL, TAP, SCREW, WHACK

Step one in the repair process is cleaning up the mess from the damaged thread, and then drilling out the hole. The existing hole should help keep the drill on track, and only light pressure should be used to enlarge the hole.

Next, you need to tap the hole you've just made, using the special tap that matches the insert you're going to use. This special tap is larger than the original thread, but will fit the outer thread on the insert.

Inserting the threaded insert requires a special tool that usually looks like a notched bar, although some manufacturers use different tapered or hexagonal keys to fit the insert. Most inserts have a bar across their end that fits into the notch in the fitting tool. With the insert in the end of the fitting tool, you simply screw the insert into the threaded hole.

Finally, you snap the bar from the end of the insert with a metal punch. Your tap repair is now complete, and you can carry on putting your project back together.

Right
The insert coil sits on the forked fitting tool with the centre bar between the tines. Once the coil is fitted, the bar is broken off with a flat-ended punch

Left
The threaded insert takes the damaged hole and restores the original 1/4" BSW thread. You can just see the insert sitting inside the hole on the vice body

YOU'LL NEED

- **V-coil** (or similar) thread repair kit for the exact size of bolt you're using
- **Tap wrench** (for using with tap and die sets, not for sink taps!)

QUICK TIP

When you screw the bolt into the threaded insert, the Heli-Coil expands slightly and locks the outer thread into place.

Wireframe

Join us as we lift the lid on video games

Visit **wfmag.cc** to learn more

TUTORIAL

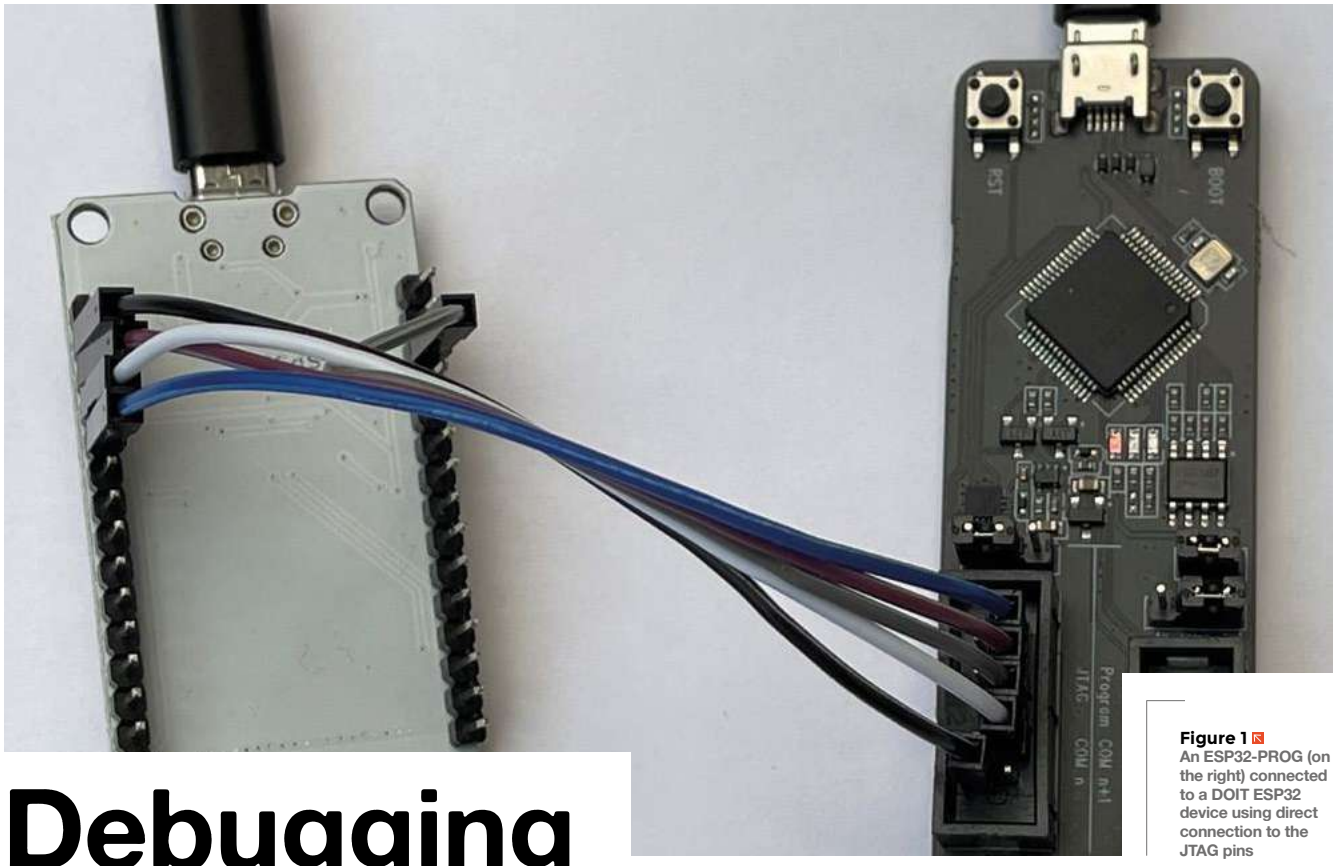


Figure 1 An ESP32-PROG (on the right) connected to a DOIT ESP32 device using direct connection to the JTAG pins

Debugging embedded software

If at first you don't succeed, debug

Rob Miles

@robmiles

Rob Miles has been playing with hardware and software since almost before there was hardware and software. You can find out more about his so-called life at **robmiles.com**.

In this article, we're going to take a look at debugging. We'll find what a debugger does and discover how to add hardware that can be used to tell us what our devices are really thinking.

BUG ORIGINS

Whenever your program doesn't do what you want it to, you've got a bug. An early bug was an insect that got stuck in the contacts of an early computer. Bugs can be caused by many things, including poor specification, programmer error, or plain bad luck. The very first programmers had no way of fixing their bugs other than staring at their code and trying to work out what had gone wrong. However, if you are writing a program on a desktop computer today, one of the tools at your disposal will be your trusty debugger. This allows you to stop a program, look at what it is doing, and then continue, or even step through individual program statements.

BUILDING CODE FOR DEBUGGING

To understand how a debugger works, we can start by considering the compilation process. Some languages, including C++, are compiled. A program called a compiler converts program source code into low-level machine code which tells the hardware what to do. This machine code is loaded into the target computer which runs your code. To see how this works, consider the following `loop` function.

```
void loop() {
  i = i + 1;
  j = j  - 1;
}
```

Address	Machine Code	Opcode	Operands	Description
83:	fd2591	l32r	a9, 40018	Make register a9 point at the location where i is stored
86:	0988	l32i.n	a8, a9, 0	Load the location pointed at by a9 into a8
88:	881b	addi.n	a8, a8, 1	Add 1 to the value in a8
8a:	0989	s32i.n	a8, a9, 0	Store a8 in the location pointed at by a9
8c:	fd2491	l32r	a9, 4001c	Make register a9 point at the location where j is stored
8f:	0988	l32i.n	a8, a9, 0	Load the location pointed at by a9 into a8
91:	880b	addi.n	a8, a8, -1	Subtract 1 from the value in a8
93:	0989	s32i.n	a8, a9, 0	Store a8 in the location pointed at by a9

Figure 2 ESP32 Assembler table

Each time the `loop` function is called, it will add 1 to the value in the variable `i` and subtract 1 from the value in the variable `j`. We might want to run this function on an ESP32 device. This can't understand C++ statements, so the compiler converts them into a sequence of instructions that it can understand.

In the **Figure 2** table, you can see the instructions produced by the compiler for the statements in the `loop` function. They have been simplified slightly and a description added. The first column shows the address in memory of that instruction. Computers store programs and data in numbered locations. When a program is running, the processor takes machine code instructions from a location and performs them. It then moves down memory to the next instruction. The instructions from the `loop` function are stored in memory starting at location number 83.

> A debugger allows you to stop a program, look **at what it is doing, and then continue**

The second column shows the machine code values stored in the ESP32. The first instruction is made up of three bytes which have the values 0xfd (in hex), 0x25, and 0x92. When the program is running, the ESP32 decodes and performs these instructions.

The opcode column contains the name of the operator, and the operands are the things that the operator works on. The opcode and operands columns aren't needed by the ESP32: it only needs the machine code bytes. Those two columns are just for us to read. From them, we can work out that the variable `i` is being held in location 40018, and the variable `j` is held in location 4001C. It is also worth noting that the ESP32 performs subtraction by adding a negative number.

GRACE HOPPER AND THE FIRST EVER 'BUG'

The word 'bug' was around well before computers. It became associated with programming when Grace Hopper, one of the world's first programmers, added the annotation 'First actual case of a bug being found' to a report of a moth that had got caught in an early computer. Grace went on to develop the world's first compiler, and she was instrumental in the development of the COBOL programming language which is still used in business today.

BREAKING IN IS HARD TO DO

If the `loop` function above is not doing what we think it should be, then we can look at the values in `i` and →

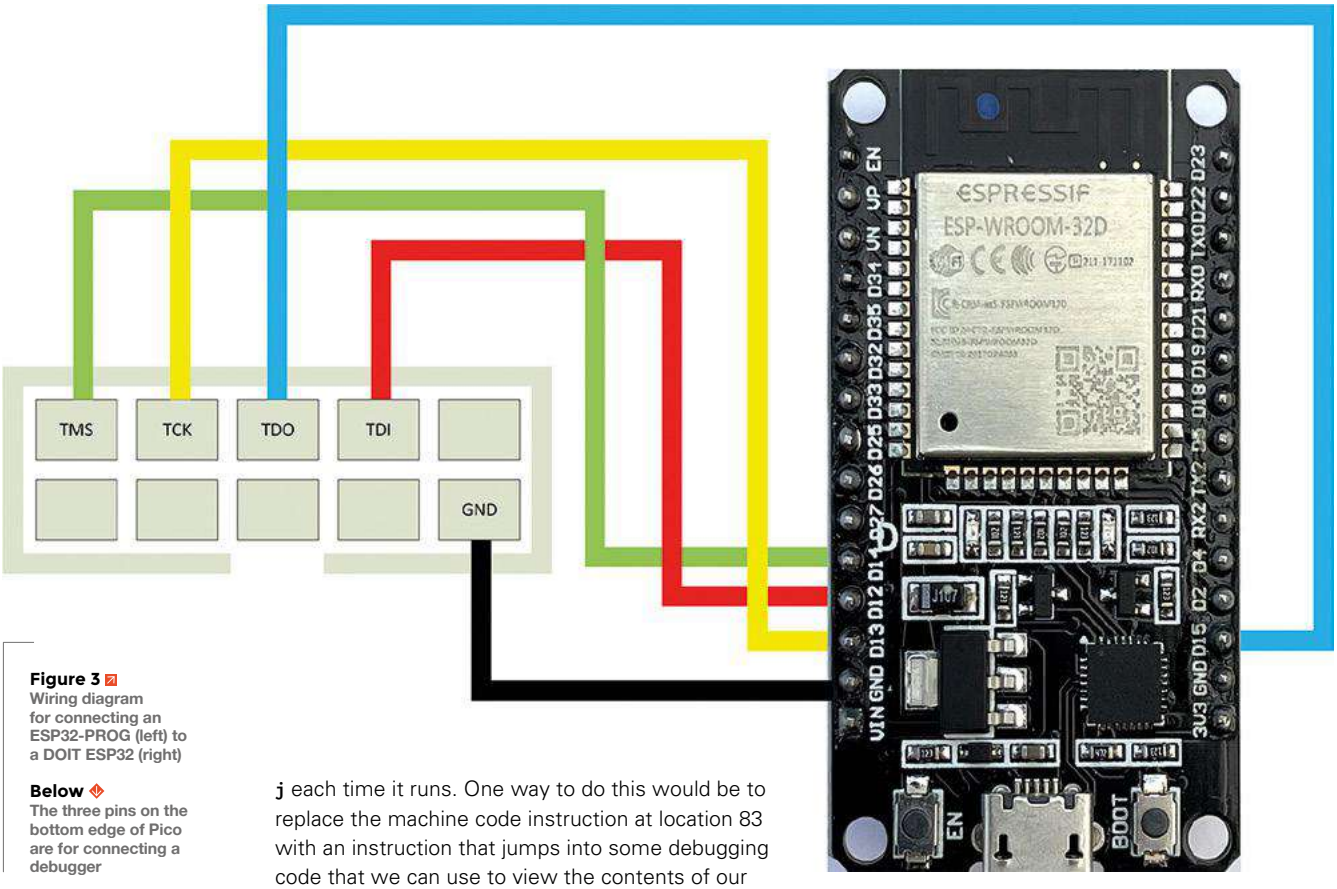
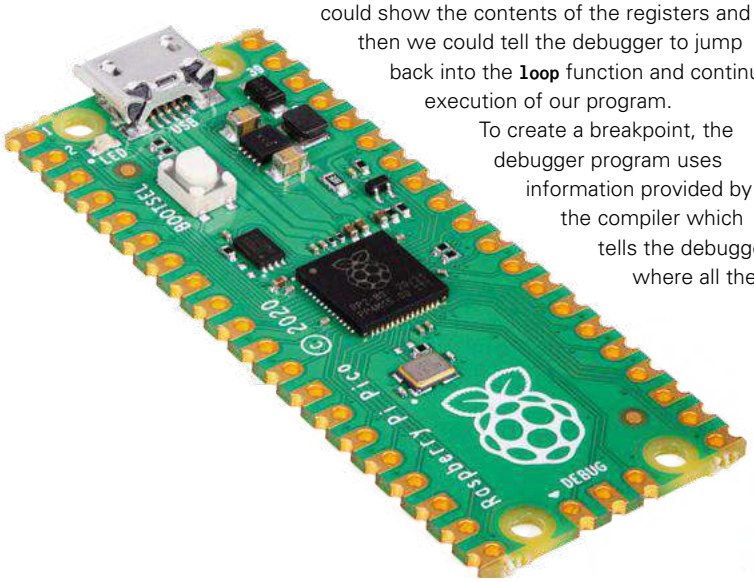


Figure 3
Wiring diagram for connecting an ESP32-PROG (left) to a DOIT ESP32 (right)

Below
The three pins on the bottom edge of Pico are for connecting a debugger

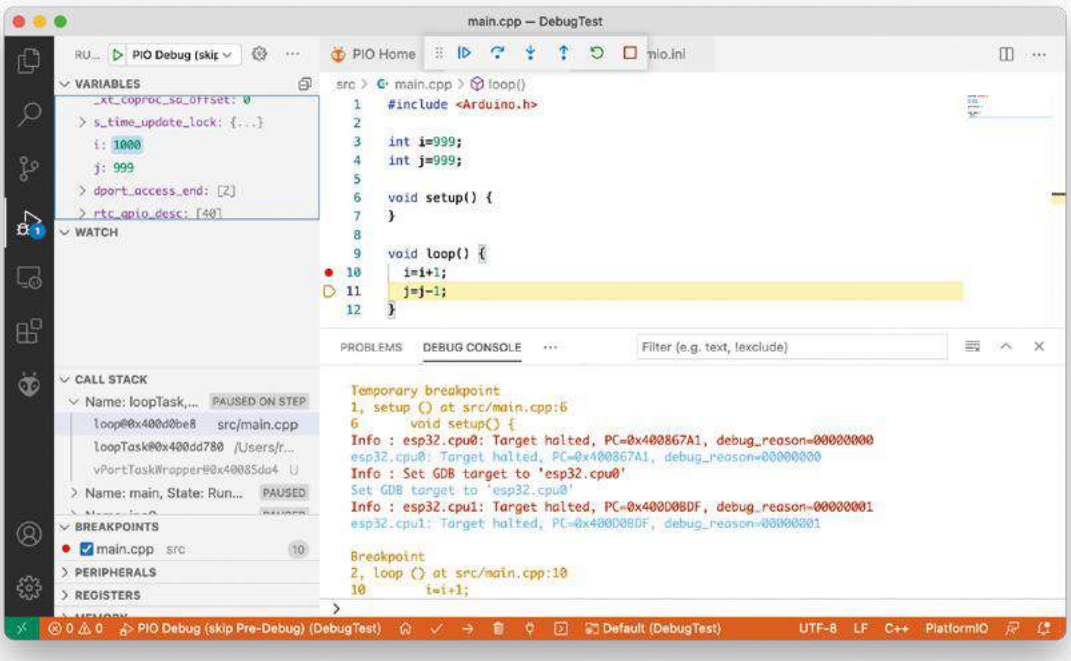


`j` each time it runs. One way to do this would be to replace the machine code instruction at location 83 with an instruction that jumps into some debugging code that we can use to view the contents of our variables. When the program reaches this statement in the program, it would enter our debugger. This is called setting a breakpoint in the code. The debugger could show the contents of the registers and then we could tell the debugger to jump back into the **`loop`** function and continue execution of our program.

To create a breakpoint, the debugger program uses information provided by the compiler which tells the debugger where all the variables are stored and the location of each program statement. This information is produced when a program is compiled in 'debug' mode. The debugger uses this to work out where to insert the breakpoint code that will pause the program. This works well if the debugger is running on the same computer as the program being debugged. However, when we are writing programs for an embedded device, this is not the case. ESP32 code is sent from our computer into the target device to run. There is no way that the debugger can set a breakpoint by modifying the program code because it doesn't have access to it. So, how can we put breakpoints into code running inside an embedded device?

WE'VE BEEN EXPECTING YOU, MR BOND

In the early days of embedded development, developers used versions of processors called 'bond-out' devices. These were special versions of processors which brought out the internal signals, including the address lines that identified the memory location that the hardware was accessing



Left
There is a breakpoint set at line 10. The program has been run to that breakpoint and then 'stepped' over the statement at line 10. The Variables section of the view shows that the value in it is now 1000, as it has been incremented by 1 by the statement at line 10

at any given instant. These chips were made by 'bonding' extra wires to the internal circuitry, hence the name.

Developers used hardware that monitored the addresses being used and detected when particular locations were being read or written. This extra hardware, called an 'in-circuit emulator', was the only way to debug early embedded code. To debug our **loop** function, we would tell the hardware to stop the device when it detected an attempt to read from the program memory at address 83 (where the machine code for the **loop** function starts). The circuitry would then read the registers in the device and allow us to view their contents. This method worked well, but the emulators were expensive and only large companies could afford them.

ENTER JTAG

As the power and complexity of microprocessors grew, it became harder to make bond-outs to expose all the internal signals that make hardware debugging possible. To address this, manufacturers formed a Joint Tag Action Group (JTAG) to define standards by which a device can expose its internal state using just a few pins. Many circuit boards have pins labelled JTAG which are used during manufacture and testing. Sometimes these pins can also be used for hardware debugging. Not all processors support hardware debugging connections. The ATmega328P processor used in the Arduino Uno cannot be debugged in this way. However, the ESP32 does provide these connections. Some of the general-purpose input/output (GPIO) pins on an ESP32 can be used as JTAG connectors. To debug code running in hardware, you'll need some way of connecting your development computer to the JTAG signals on the target device. Espressif (the same company that makes the ESP32) produces a great device for this. It is called the ESP32-PROG.

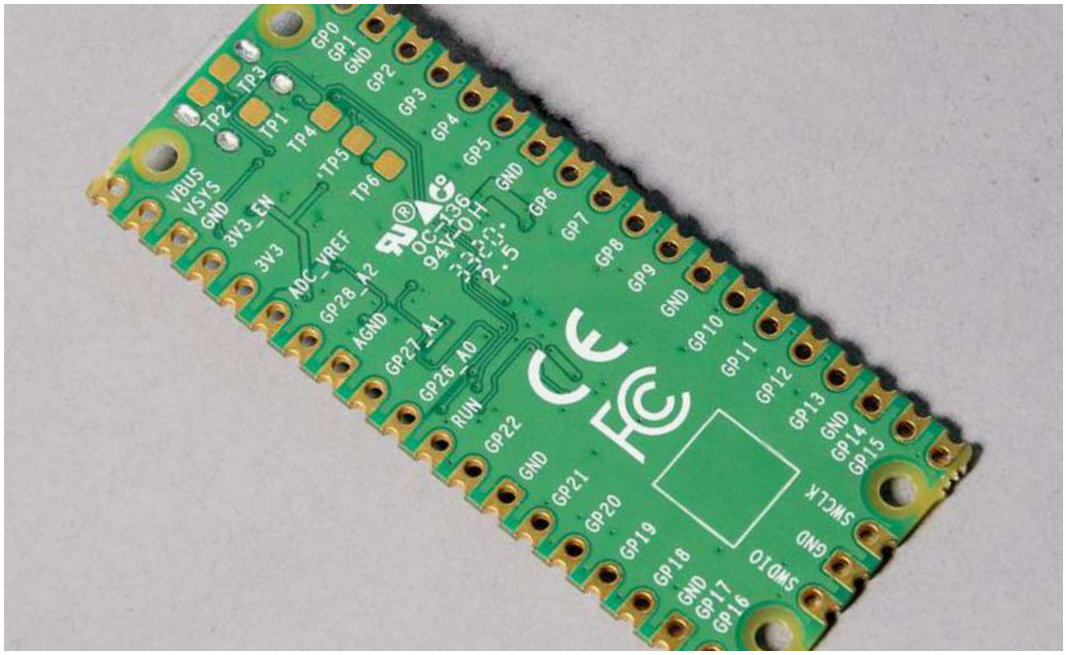
The ESP32-PROG

You can pick up an ESP32-PROG device for around £15 or so. It can also be used to program an ESP32. It can be connected via a ribbon cable or you can use DuPont cables (socket to socket), as shown in **Figure 1**.

TDI (Test Data In)	ESP32 GPIO12
TDO (Test Data Out)	ESP32 GPIO15
TCK (Test Clock)	ESP32 GPIO13
TMS (Test Mode Select)	ESP32 GPIO14

The table above shows the connections between an ESP32 and the ESP-PROG device.

Figure 3 shows how the socket on the ESP-PROG can be connected to an ESP32 device. Note that both the ESP32 and the ESP-PROG will need to be connected to a power source via their micro USB →



Right
SWD stands for Serial Wire Debug

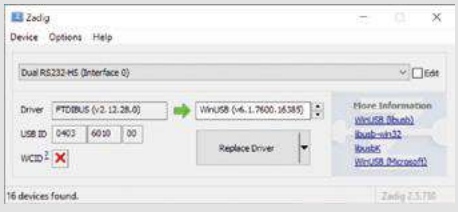
connectors. You will still deploy your program using a connection to the ESP32 device. If you encounter problems with program deployment, disconnect the ESP32 USB cable from your PC and try again.

THE OPENOCD CONNECTION

The debugging itself is managed by the 'Open On-Chip Debugger' (OpenOCD) software. This provides a connection between the hardware and the software environment that you use to write and debug your code. OpenOCD talks to the ESP-PROG device over USB. The ESP-PROG provides two serial port connections to the host computer. One can be used for programming an ESP32 via the 6-pin connector on the ESP-PROG. The other is used to control debugging.

WINDOWS USB CONFIGURATION

Unfortunately, when you first plug an ESP-PROG device into a Windows PC, it selects USB drivers that will not work with OpenOCD. If you are using a Windows 10 PC, you will have to use a tool called Zadig to install the correct ones. Download it from **zadig.akeo.ie**.



Above
Run the program, select Options > List All Devices, and then change the driver for Dual RS232-HS (Interface 0) to the one shown here

DEBUGGING WITH VISUAL STUDIO CODE AND PLATFORMIO

Visual Studio Code is a free development environment that runs on PC, Mac, and Raspberry Pi. PlatformIO is a free plug-in for embedded development using Visual Studio Code. PlatformIO includes the OpenOCD framework. A PlatformIO project contains a **platform.ini** file that contains the project configuration options. We need to edit this file and add two lines to our configuration:

```
debug_tool = esp-prog
debug_init_break = tbreak setup
```

Now we can open up the debug window in Visual Studio Code and start the debugger.

HARDWARE DEBUGGING WITH RASPBERRY PI PICO

The Raspberry Pi Pico device exposes JTAG signals that can be used for hardware debugging. You can wire these directly to a Raspberry Pi and use that as the debugging and development platform, or you can use another Raspberry Pi Pico device as a debugging probe.

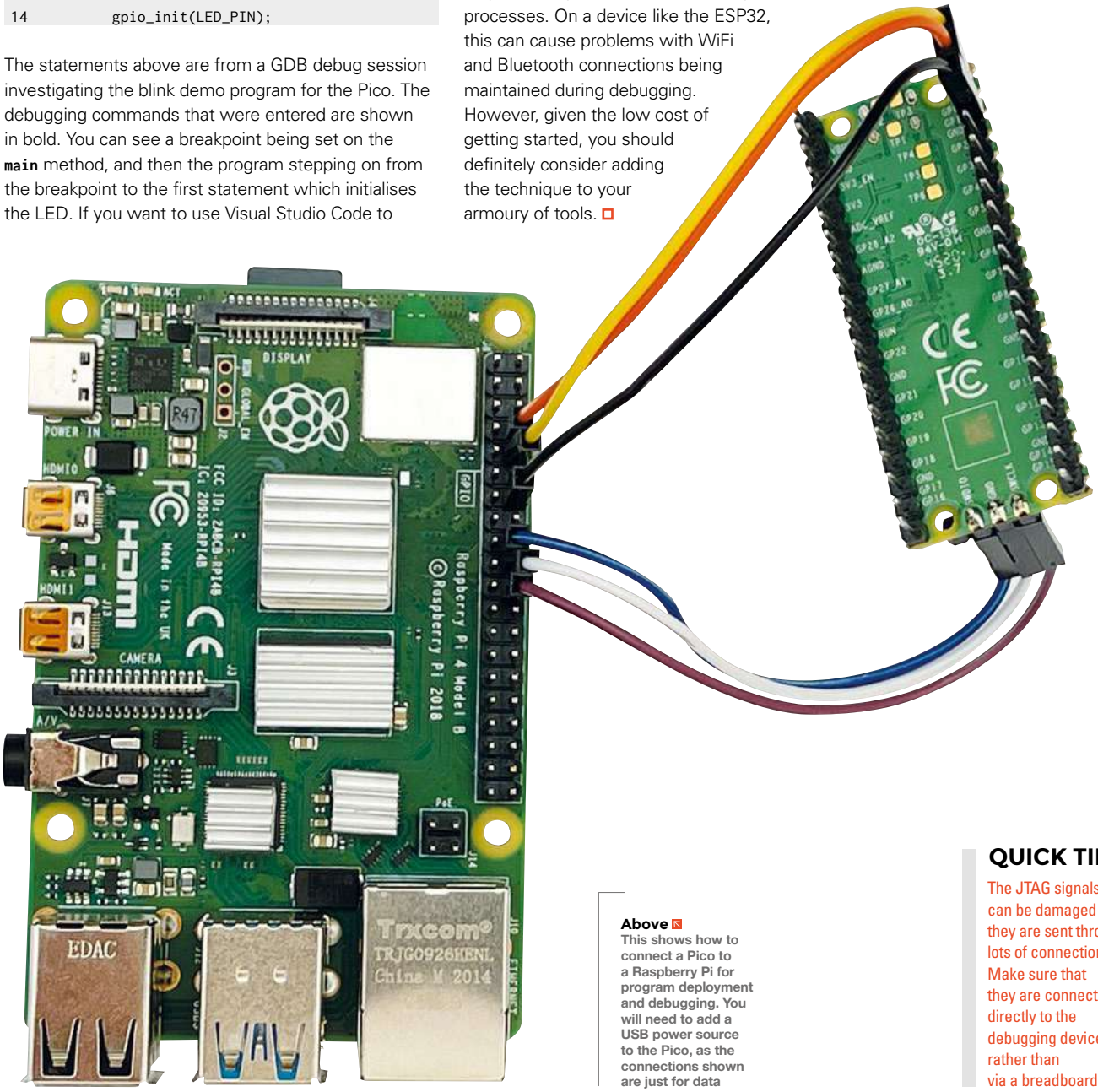
The Pico documentation gives detailed instructions on how to do this here: **hsmag.cc/PicoDatasheets**. You can use the GNU Debugger to debug a program from the command line.

```
(gdb) b main
Breakpoint 1 at 0x1000035c: file /home/pi/pico/
pico-examples/blink/blink.c, line 9.
(gdb) continue
Continuing.
Thread 1 hit Breakpoint 1, main () at /home/pi/
pico/pico-examples/blink/blink.c:9
9       int main() {
(gdb) step
14          gpio_init(LED_PIN);
```

The statements above are from a GDB debug session investigating the blink demo program for the Pico. The debugging commands that were entered are shown in bold. You can see a breakpoint being set on the **main** method, and then the program stepping on from the breakpoint to the first statement which initialises the LED. If you want to use Visual Studio Code to debug your programs on Raspberry Pi, you can do this as well.

HARDWARE DEBUGGING FOR THE WIN

Hardware debugging is very powerful. It lets you look inside your devices to see exactly what they are doing. You do need to be a bit careful when you use it sometimes, because the debugging process stops the target device and all background processes. On a device like the ESP32, this can cause problems with WiFi and Bluetooth connections being maintained during debugging. However, given the low cost of getting started, you should definitely consider adding the technique to your armoury of tools.



Above
This shows how to connect a Pico to a Raspberry Pi for program deployment and debugging. You will need to add a USB power source to the Pico, as the connections shown are just for data

QUICK TIP

The JTAG signals can be damaged if they are sent through lots of connections. Make sure that they are connected directly to the debugging device rather than via a breadboard.

HackSpace
TECHNOLOGY IN YOUR HANDS

FIELD TEST

HACK | MAKE | BUILD | CREATE

Hacker gear poked, prodded, taken apart, and investigated

ONLY THE BEST

Soldering irons and stations

How to pick the right tool for your workbench

By **Marc de Vinck** @devinck

When getting into electronics as a hobby or a profession, the first tool that comes to mind is the soldering iron. Sure, there are now solderless kits, many of which I love, but the reality is, as you progress past being a beginner, you'll need a soldering iron. And picking your first soldering iron is often done incorrectly, and results in a bad experience.

So how can you pick the wrong iron? It's easy to make a mistake. Typically, I see one of two things. The first, becoming less common nowadays, is the 'I used this gun-looking iron I found in the garage and it didn't work'. Yep, you got that right. Those pistol grip types of irons are old school and not appropriate for modern electronics. Avoid them at all costs.

The other, far more common response, is 'I picked up a stick-like iron for a couple of dollars and it didn't work'. Yeah, they do work, but they are also known to many of us as 'fire sticks'. Why? Because they tend to catch fire! And more importantly, they rarely have temperature adjustment capabilities. If your iron can't change temperatures, don't use it. It's not that you really need to change the temperature much, but if you can't, it will also most likely not be adjusted to the proper temperature, or hold the temperature properly. Oh, and did I mention the cheap ones tend to eventually catch fire?

In this Best of Breed, we'll go over a bunch of great picks for your first iron, plus we have a few extra items that you'll want to pick up too! It's not a comprehensive list of tools or irons, but it will get you familiarised with what to look for when selecting your soldering iron.

Hakko FX-888D vs Weller WE1010 Soldering Station

HAKKO $129.95 | adafruit.com

WELLER $129.00 | sparkfun.com

This iron is my personal go-to iron. I've had it for a few years, and it never fails to perform. I rarely change the temperature, but I do appreciate the digital display for keeping track of when it's hot enough to start working. But the biggest feature that I think is often overlooked is the stand. It's robust, heavy, and keeps your iron safely secured when hot and not in use. I also insist on using a brass sponge instead of a wet sponge, as it works better and keeps the temperature consistent when working. Fortunately, this iron has both!

The Hakko line of irons are known for their reliability and variety of interchangeable tips. This model has a temperature range of 120 °F – 899 °F (50 °C – 480 °C), has 65 W power, and has a temperature stability of ±1.8 °F (1.0 °C). Whether you are a beginner or a professional, you really can't go wrong with this iron.

Below
The fluid design of the Hakko stands out against the other more box-like designs

Above
Almost everything you could want in a soldering iron

Another great brand when it comes to reliable and well-known soldering irons is Weller. The WE1010 is a powerful 70 W soldering station that is a good choice for anyone getting started in soldering electronics. And just like the Hakko, it features a digital readout and fine temperature adjustment, making soldering so easy.

The iron heats up incredibly fast, is ESD-safe, and you can pick up interchangeable tips for a variety of different soldering tasks. This is close to a perfect soldering station, except for the fact that the spring-type stand isn't that great, and only includes a sponge to clean the tip of the iron. But, if you are willing to add a brass sponge to your kit, then this is a solid choice. →

VERDICT

Hakko FX-888D

Can't go wrong with a Hakko.

10/10

Weller WE1010 Soldering Station

So close to perfect.

9/10

BEST OF BREED

TS80 USB C Powered Soldering Iron

TS80 ◈ **$109.95** | **adafruit.com**

I've seen USB-powered soldering irons in the past, and they have all been nothing more than a novelty item. But the TS80 is a bit different. This iron features a USB-C connector and power adapter that allows it to run at 9V and 2A. Now you can get a fully adjustable temperature range from 100°C to 400°C in a slim, pen-style soldering iron.

The iron also features an OLED screen and has auto power off. Just note that you can't plug this into a computer's USB port and expect 9V and 2A, so you must use the included power supply. So, it's USB, but not necessarily what you first think. In any case, having a slim, pen-style iron with a flexible cord is really nice.

VERDICT

TS80 USB C Powered Soldering Iron

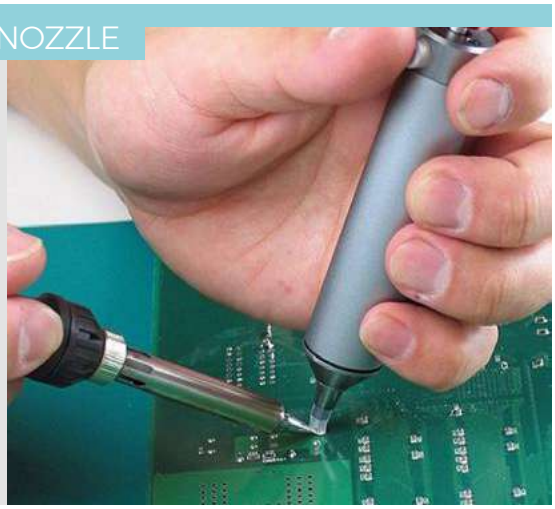
A great portable solution.

8/10

SOLDER SUCKER WITH SILICONE NOZZLE

PIMORONI ◈ **$21.39** | **pimoroni.com**

At some point you'll make a mistake, and you only have a few choices when it comes to fixing those mistakes. You can start over, use solder braid, or use my favourite accessory – the solder sucker. Using a solder sucker to remove hot solder can be tricky. However, if you purchase one like this one from Pimoroni, you'll have a lot more success. The secret is a little silicone tubing at the tip. It won't melt, and it allows you to get really close to the molten solder before pressing the button, sucking up all the solder. If you haven't used one of these types of solder suckers before, you're in for a nice surprise. They are great!



Left ◈
Because we all make mistakes

Weller WLC100 Soldering Station

WELLER ◈ **$44.95** | **sparkfun.com**

For anyone on a tight budget, the Weller WLC100 Soldering Station from SparkFun is a good choice. At around half the price of most other soldering stations, the Weller is great value and also a trusted name in electronics. You still get an adjustable temperature – just like the more expensive versions – but it's a rotary dial, so the actual temperature is unknown. That really isn't a big deal for beginners, as once you find the right setting, you really won't change it much.

The Weller WLC100 Soldering Station's power ranges from 5 W to 40 W, with a maximum temperature of 900 °F (482 °C), which is powerful enough for almost any type of soldering work. If you have a limited budget, this is an ideal choice – it's far better than any non-temperature adjustable irons out there. →

Left ◈ The WLC100 is compact by soldering station standards

VERDICT

Weller WLC100 Soldering Station

A solid choice for your first iron.

8/10

MINI HOT PLATE PREHEATER WITH USB-C POWER SUPPLY

ADAFRUIT ◈ **$99.95** | **adafruit.com**

Sometimes a soldering iron just isn't the right tool for the job. After you learn to solder through-hole components, you'll often find yourself soldering surface mount components. And although you can solder them by hand, a reflow oven or hotplate is much easier and more reliable. They can be big and expensive, but not this little hotplate from Adafruit. I know a few people who have this handy little hotplate, and it works great. It's perfect for rework, preheating, or even reflowing solder paste. Just make sure your PCB is small because it's only 30 mm × 30 mm.

BEST OF BREED

Hot-Air Rework Station - 303D

SPARKFUN $129.95 | sparkfun.com

Once you have a decent soldering iron or soldering station, you'll eventually find yourself wanting a little more versatility when it comes to soldering and reworking solder connections. And that's when you'll be looking for a Hot-Air Rework Station. The 303D from SparkFun features tight temperature tolerances and large airflows. It also has a digital readout of the actual air temperature and airflow rate, up to 23 litres per minute. It's perfect for surface mount reflow, repair, or removal of components. You can also use it for shrink-wrapping and thermoplastic welding.

Of course, you can solder many surface mount components by hand with a traditional soldering iron, but making repairs or scavenging parts from a PCB with this type of station makes it so much easier. There are also plenty of surface mount electronic components that can only be soldered with a reflow oven or rework station. □

Below
Surface mount work can be easier with hot air

VERDICT

Hot-Air Rework Station - 303D

When you need it, you need it.

9/10

DELUXE SOLDERING IRON STAND WITH SOLDER ROLL HOLDER

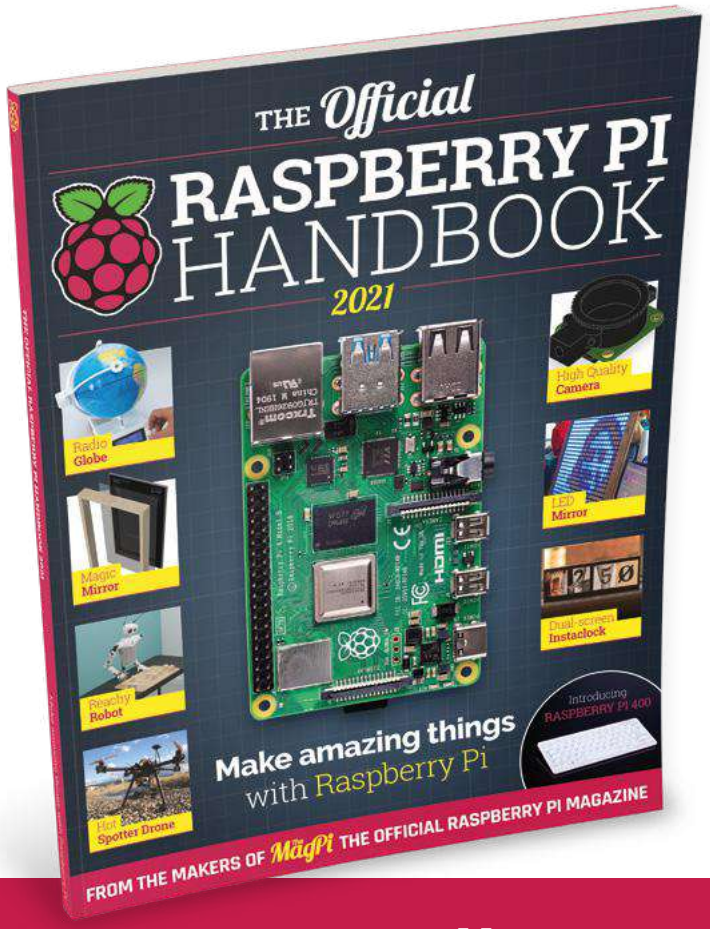
JAMECO $21.95 | jameco.com

One thing that is often overlooked when purchasing a soldering iron is the stand. You really want a heavy and secure stand for your iron. It makes soldering safer and more convenient. I like this deluxe stand because it also has a solder holder, which is another item often overlooked. If this only had a brass sponge, it would be the perfect stand.

THE *Official* RASPBERRY PI HANDBOOK 2021

200 PAGES OF RASPBERRY PI

- Get started with Raspberry Pi, electronics, and more
- Be inspired by incredible projects made by other people
- Learn how to code and make with our step-by-step tutorials
- Find out about the top kits and accessories for your projects

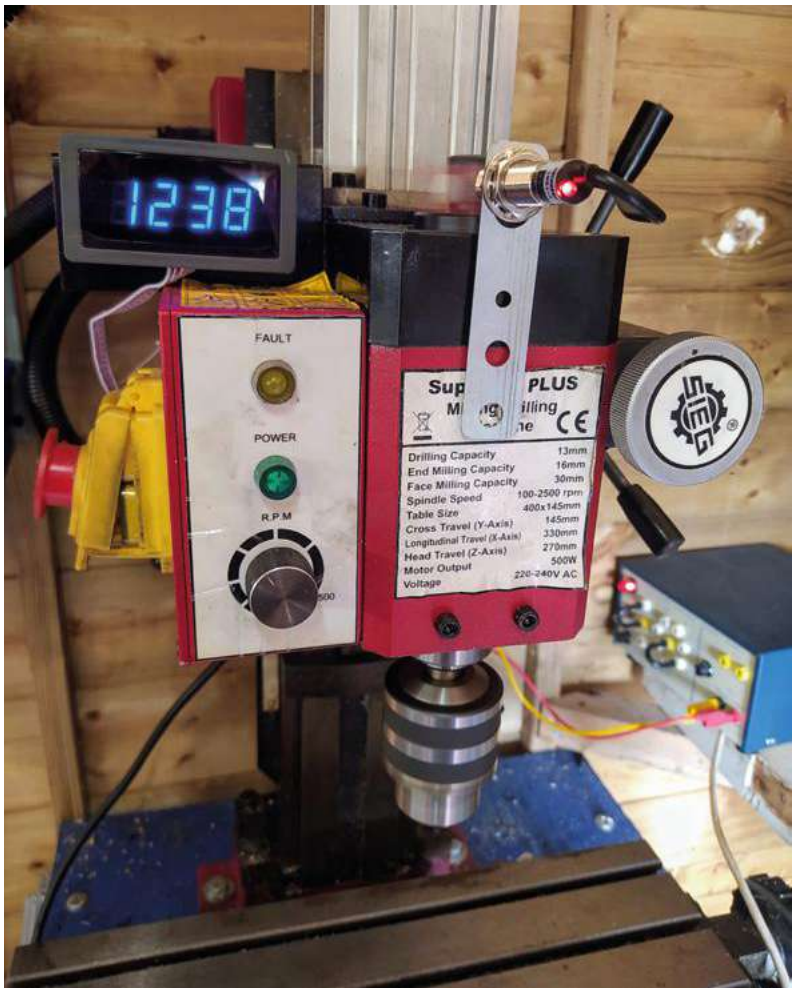


Buy online: magpi.cc/store

Tachometer RPM kit

Find out how fast your tools are spinning

By Jo Hinchliffe @concreted0g



Below The tachometer kit fitted to our milling machine spindle, giving precise results on the rpm of the variable-speed spindle

Often tools, machines, and projects that we might build have something that spins, and we would like to know the number of revolutions per minute. There's a range of cheap solutions to this, from the mechanical to a hand-held IR gun. For a permanent installation like on a pillar drill or a milling machine, however, a common kit is a tachometer comprising a Hall effect switch and an LED display.

These kits are sold on a variety of online platforms and described with a variety of labels, from 'Four LED Digital Tachometer' to 'Digital Speed Meter'. They are available for a range of prices, from £9 to £16. Ours cost £11, and the box contained the four-digit LED display, a short ribbon cable with a JST connector, the Hall effect switch/sensor, a mounting plate for the sensor, and a small magnet. Of note, ours arrived with absolutely no instructions or any form of wiring diagram!

The idea is pretty straightforward – the device is powered by an 8–24 V supply (not supplied with the kit). We used a 12 V power supply that we had spare. There is a Hall effect sensor switch which you mount near to your spinning object. In our case, we mounted it temporarily to the top of our milling machine. In turn, the magnet supplied is stuck to the spinning object in a position where it will pass the sensor. The sensor and magnet need to pass reasonably closely, and we found that if the distance was more than around 10 mm, the device wouldn't register and the display would show zeros.

After some digging around online, it seems there are two versions of this device in terms of wiring,

Left
A very quick 3D print job to make it easier to mount the magnet on our milling machine

Below
Our sensor setup with the magnet attached to the spindle, and the Hall effect sensor switch temporarily attached using some spare magnets

with some having a four-wire connector on the display unit and some having a five-wire connector. It also appears that it's common for either type to be supplied without any wiring diagram! Ours was the five-wire version and, with some more searching online, we managed to wire it up.

We wanted to fit this to our milling machine spindle, which reportedly can be adjusted between 100 rpm and 2500 rpm. We mounted it at the top of the spindle, which is out of the way but complicated by the fact that our milling machine spindle has a draw-bar that regularly needs to be removed with a spanner and a sharp tap with a hammer, making it unsuitable to glue the magnet to. We ended up 3D-printing a small magnet holder, which we then fixed to the spindle retaining lock nuts with a small dab of epoxy, with the magnet fixed into the 3D print with a spot of superglue. There's a supplied metal mount with some fixing holes for the Hall sensor, and you can adjust the length that the sensor sticks out from the mount. We will probably make a small 3D print to affix the sensor onto the side of the spindle housing but, for these tests, we found that we could mount the sensor to the front of the spindle housing using a couple of strong magnets.

Making sure that the magnet housing was firmly affixed and with everything wired up, we turned on the 12 V power supply. Available in red, green, or blue, we had opted for the blue LED display, and we were pleased to see how bright and clear this is, certainly readable from a distance. Speaking of the display, the plastic housing for it has wedge-type spring clips built into it, meaning that it could be mounted into a panel with the correct size rectangular cut-out. At some point, we will probably make an enclosure and some form of conduit or sleeving for the wiring to protect it all from hot metal chips and coolant/cutting fluid.

> "The unit itself performed **excellently, and we think it is pretty accurate**"

The unit itself performed excellently, and we think it is pretty accurate: our brushless Sieg SX2P machine does indeed max out at exactly 2500 rpm. Readings take a few moments to settle when you change the speed, and it can flicker a couple of rpm at times. It's certainly an improvement over our previous 'guess the speed of the spindle' approach. We can imagine, particularly at this price, adding them to other machines in our collection. ◻

DIRECT FROM **SHENZHEN**

MonkMakes Air Quality Kit

Make sure you stay healthy with fresh air

MONKMAKES ◈ **£42** | **monkmakes.com**

By **Ben Everard** @ben_everard

When you hear about rising CO_2 levels, you might think of global warming – and of course, this is a huge issue. However, CO_2 can cause problems on a much smaller scale. People exhale more CO_2 than they inhale, so if you have a lot of people in an enclosed space, the CO_2 levels can rise dramatically, and this can cause foggy-headedness and cognitive impairment.

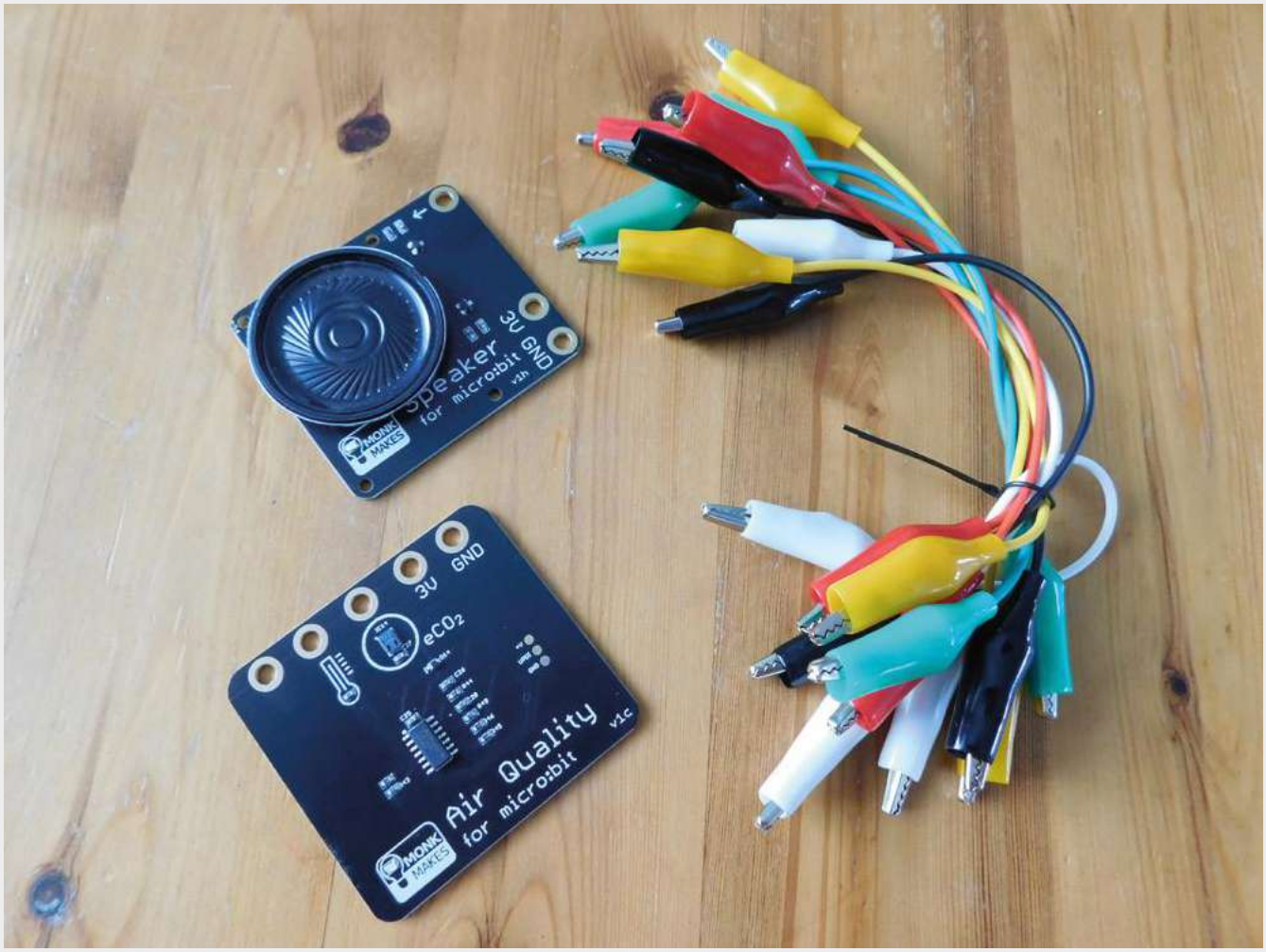
Below ◈
The chunky square boards are easy to work with – even for young makers with small hands

Getting fresh air through an open window (or other ventilation) is vital to help us think and work to our full capabilities. If levels of CO_2 are building up indoors, then it's a pretty good indicator that other things that people exhale are also building up, such as viruses that will not be named here.

This kit, from MonkMakes, adds an eCO2 sensor to your micro:bit and a speaker that can chirp or sound an alarm if it should get too high. Almost all consumer-grade CO_2 sensors are actually eCO_2 sensors, and the 'e' stands for estimated. It's actually quite hard to electrically measure CO_2, so instead, these sensors measure volatile organic compounds (VOCs), which people also exhale. From the level of VOCs, these devices estimate the CO_2 level in the air. It's not perfect, but for our purposes, it's a pretty good proxy, especially because we should probably get more ventilation if there's a large amount of VOCs, regardless of CO_2 levels. If you want true CO_2 readings, MonkMakes does offer a CO_2 sensor, but it's much more expensive than this kit (£120).

The air quality board has two outputs: one for temperature and one for eCO_2. In both cases, these output an analogue voltage that you just need to read through micro:bit analogue input. There's nothing unique about the micro:bit's analogue input, and you could equally pair this with any other 3V board that has a suitable input (such as Raspberry Pi Pico). However, the code examples in the instructions will only work with micro:bit.

This analogue output makes it very easy to use, but you do sacrifice a little accuracy. Long cables and poor crocodile-clip joints may affect the readings, but this need not



trouble you too much for most cases. You should easily get enough accuracy to give you an idea of what's going on in the air, and giving you a reminder to open a window.

While the sensing board might be the star of the show, we're fans of the MonkMakes Speaker board. It makes it really easy to add audio to a whole range of projects – not just micro:bit, but any 3V microcontroller. It includes an amplifier and speaker in one, so there's no wiring to do: just hook it up to power and a signal and it'll do the rest. It's quite small, so not suitable if you want to crank the volume up, but great for little projects where you want a beep and don't want to use headphones.

Alongside the hardware, you get a 24-page booklet with a Getting Started guide, four projects, and a trouble-shooting section. It's this documentation that raises this from a slightly obscure product to an interesting beginner-level kit, to help people get a better understanding of both programming and air quality. As well as the basics of how to read the sensor, these projects explore the micro:bit's abilities, such as showing data in the LED grid, and even sending data to a web browser to collate the results for later processing. This last example shows how this kit can very easily be used as a data logger.

> "You should easily get enough accuracy to give you **an idea of what's going on in the air**"

You can work through the projects in either the block-based MakeCode web-based environment or MicroPython, and they'll work with both micro:bit v1 and v2.

The hardware in this kit is really easy to use. Connecting it up is just a case of clipping some crocodile clips on the appropriate bits. Programming it is also about as easy as it can be. It gives you the ability to turn your micro:bit into a useful environmental sensor, while learning a little about programming along the way. If you're just getting started with microcontrollers, or looking to build an air quality sensor, then this is a great option. ◻

Above
The included crocodile clips are quite short, which helps keep everything tidy

VERDICT

Documentation makes this a great kit for beginners.

9/10

next month

issue #44 ON SALE 17 JUNE

ARDUINO

ALSO

- 3D PRINTING
- CIRCUITPYTHON
- RASPBERRY PI
- MAKING MUSIC
- AND MUCH MORE

DON'T MISS OUT

hsmag.cc/subscribe

“MIDI Fighter-style controllers (MIDI controllers with grids of arcade buttons) have been a staple of the DIY MIDI controller community for years. This project continues that tradition with the Raspberry Pi Pico. A grid of 16 arcade buttons lets you play MIDI notes faster than you can yell “Hadouken!”, either live with hardware or with your digital audio workstation (DAW) of choice.”

See page 52 for how Liz Clark built this.

Board?

FREE SHIPPING
ON ORDERS OVER £33 OR $50 USD*

0800 587 0991
DIGIKEY.CO.UK

Digi-Key® ELECTRONICS

10 MILLION+ PRODUCTS ONLINE | 1,300+ INDUSTRY-LEADING SUPPLIERS | 100% FRANCHISED DISTRIBUTOR

*A shipping charge of £12.00 will be billed on all orders of less than £33.00. A shipping charge of $18.00 USD will be billed on all orders of less than $50.00 USD. All orders are shipped via UPS, Federal Express, or DHL for delivery within 1-3 days (dependent on final destination). No handling fees. All prices are in British pound sterling or United States dollar. Digi-Key is a franchised distributor for all supplier partners. New products added daily. Digi-Key and Digi-Key Electronics are registered trademarks of Digi-Key Electronics in the U.S. and other countries.

ECIA MEMBER
Supporting The Authorized Channel